DOCUMENT RESUME

ED 382 647 TM 023 081

AUTHOR Thomas, Neal
TITLE The E-Step of the MGROUP EM Algorithm. Program Statistics Research Technical Report No. 93-37.
INSTITUTION Educational Testing Service, Princeton, N.J.
REPORT NO ETS-RR-93-54
PUB DATE Nov 93
NOTE 77p.
PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS Algorithms; Elementary Secondary Education; *Estimation (Mathematics); Evaluation Methods; *Item Response Theory; Models; *Statistical Distributions
IDENTIFIERS Asymptotic Distributions; *EM Algorithm; *National Assessment of Educational Progress

ABSTRACT

Mislevy (1984, 1985) introduced an EM algorithm for estimating the parameters of a latent distribution model that is used extensively by the National Assessment of Educational Progress. Second order asymptotic corrections are derived and applied along with more common first order asymptotic corrections to approximate the expectations required by the E-step of the EM algorithm. These corrections produce much more accurate approximations than those produced by two different normal approximations. The asymptotic corrections are applicable in high dimensional models. The number of function evaluations required by these methods grows linearly as dimension increases, in contrast to the exponential growth with dimension of product quadrature rules which yield similar accuracy. Two figures and two tables illustrate the analysis. Six appendixes present various subroutines. (Contains 22 references.)
(Author/SLD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 382 647

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

R. Coley

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

RR-93-54

The E-Step of the MGROUP EM Algorithm

Neal Thomas
Educational Testing Service

PROGRAM
STATISTICS
RESEARCH

Technical Report No. 93-37

Educational Testing Service
Princeton, New Jersey 08541

TM023081

BEST COPY AVAILABLE

The Program Statistics Research Technical Report Series is designed to make the working papers of the Research Statistics Group at Educational Testing Service generally available. The series consists of reports by the members of the Research Statistics Group as well as their external and visiting statistical consultants.

Reproduction of any portion of a Program Statistics Research Technical Report requires the written consent of the author(s).

The E-Step of the MGROUP EM Algorithm

Neal Thomas
Educational Testing Service

Program Statistics Research
Technical Report No. 93-37

Research Report No. 93-54

Educational Testing Service
Princeton, NewJersey 08541

November 1993

Copyright © 1993 by Educational Testing Service. All rights reserved.

Preface

This technical report is an expansion of "Higher order asymptotic corrections applied in an EM algorithm for estimating educational proficiencies". It includes a brief overview of the MGROUP computer program originally written by Robert Mislevy and Kathy Sheehan. The primary purpose of this report is to provide detailed documentation of the E-step calculations implemented for the 1992 NAEP assessments. This report does not describe the input files and commands required to execute the MGROUP program.

Abstract

Mislevy(1984, 1985) introduced an EM algorithm for estimating the parameters of a latent distribution model that is used extensively by the National Assessment of Educational Progress. Second order asymptotic corrections are derived and applied along with more common first order asymptotic corrections to approximate the expectations required by the E-step of the EM algorithm. These corrections produce much more accurate approximations than those produced by two different normal approximations. The asymptotic corrections are applicable in high dimensional models. The number of function evaluations required by these methods grows linearly as dimension increases, in contrast to the exponential growth with dimension of product quadrature rules which yield similar accuracy.

Keywords and phrases. EM algorithm, higher order asymptotic corrections, Laplace expansion, item response models.

Acknowledgments. Paul Holland, Gene Johnson, Charles Lewis, John Mazzeo, Bob Mislevy, and Ming mei Wang made many helpful comments and suggestions for this paper. Al Rogers and Dorothy Thayer assisted in much of the computing.

1 Introduction

A multivariate multiple regression model with p latent variables representing proficiency in several closely related subject areas is used extensively to summarize educational testing data obtained from national surveys. For a sample of n examinees, the proficiency variables, denoted by $\boldsymbol{\theta}_i' = (\theta_{i1}, \ldots, \theta_{ip})$, $i = 1, \ldots, n$, are assumed to follow a multivariate normal distribution, $\boldsymbol{\theta}_i \sim N(\boldsymbol{\Gamma}'\boldsymbol{x}_i, \boldsymbol{\Sigma})$, where $\boldsymbol{x}_i' = (x_{i1}, \ldots, x_{im})$, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$, are m fully measured demographic and educational characteristics for each examinee, $\boldsymbol{\Gamma} = \left[\gamma_1 \mid \ldots \mid \gamma_p\right]$ are the $m * p$ parameters of the regression of $\boldsymbol{\theta}$ on $\boldsymbol{x}$, and $\boldsymbol{\Sigma}$ is the variance-covariance matrix for $\boldsymbol{\theta}$ conditional on $\boldsymbol{x}$. The $(\theta_1, \ldots, \theta_p)$ are often referred to as *scales* in the psychometric literature. For example, several large mathematics assessments use the scales *numbers and operations*, *measurement*, *geometry*, and *algebra*. This model is used extensively for scaling, equating, generation of multiple imputations, and other reporting purposes by the National Assessment of Educational Progress (NAEP), (Johnson and Zwick, 1990, Mislevy, 1991, Mislevy, Johnson, and Muraki, 1992), and the Adult Literacy Survey, (Kirsch and Jungeblut, 1986). An overview of the statistical methods used in NAEP is given in Beaton and Zwick(1992).

The variable $\boldsymbol{\theta}_i$ is not measured directly, but information about $\boldsymbol{\theta}_i$ is obtained from several examination items given to each examinee denoted by $\boldsymbol{y}_i$, $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$. Conditional on the $\boldsymbol{\theta}_i$, the responses of different examinees are assumed to be independent with a distribution denoted by $\prod_{i=1}^{n} f(\boldsymbol{y}_i \mid \boldsymbol{\theta}_i)$. The key feature used to form improved asymptotic approximations is that each examination item contributes information about a single scale. Decomposing the responses to the examination items for the i^{th} examinee as $\boldsymbol{y}_i = \left(\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{ip}\right)$, corresponding to the scales $(\theta_{i1}, \ldots, \theta_{ip})$ respectively, $f(\boldsymbol{y}_i \mid \boldsymbol{\theta}_i)$ can be written in the form

$$f(\boldsymbol{y}_i \mid \boldsymbol{\theta}_i) = f_1(\boldsymbol{y}_{i1} \mid \theta_{i1}) \cdots f_p\left(\boldsymbol{y}_{ip} \mid \theta_{ip}\right) \equiv l_1(\theta_1) \cdots l_p(\theta_p) \equiv l(\boldsymbol{\theta}) \ . \tag{1.1}$$

Conditional on the $\boldsymbol{\theta}_i$, the responses to the examination items measuring the same proficiency scale of an examinee are also assumed to be independent, so that $f_j\left(\boldsymbol{y}_{ij}\right)$ is

the product of the response probabilities for each examination item administered. For dichotomous items represented by $y = 0, 1$, a three parameter logistic item response model is used,

$$P(y = 1 \mid \theta) = \gamma + (1 - \gamma) / \{1 + \exp[\alpha(\theta - \beta)]\} , \tag{1.2}$$

and for ordered polytomously scored items, $y = 1, \ldots, c$, a generalized partial credit model is used,

$$P(y = k \mid \theta) = \frac{\exp\left[\sum_{j=1}^{k} \alpha(\theta - \beta_j)\right]}{\sum_{l=1}^{c} \exp\left[\sum_{j=1}^{l} \alpha(\theta - \beta_j)\right]} , \tag{1.3}$$

where $\beta_1 \equiv 0$. Hambleton and Swaminathan(1985), and Bock and Aitkin(1981) describe the model and the estimation of the item specific parameters (α, β, γ) for the dichotomous responses, and Muraki(1992) describes the polytomous response model.

With the item specific parameters estimated separately and regarded as known, the likelihood function for $(\boldsymbol{\Gamma}, \boldsymbol{\Sigma})$ based on the $(\boldsymbol{X}, \boldsymbol{Y})$ data is given by

$$\prod_{i=1}^{n} \int f_1(y_{i1} \mid \theta_{i1}) \cdots f_p\left(y_{ip} \mid \theta_{ip}\right) N(\boldsymbol{\theta}_i;\ \boldsymbol{\Gamma}' \boldsymbol{x}_i, \boldsymbol{\Sigma})\, d\boldsymbol{\theta}_i . \tag{1.4}$$

The function in (1.1) is called the examinee likelihood function because it is the likelihood function for estimating $\boldsymbol{\theta}_i$ from the $\boldsymbol{y}_i$ data, and the integrand in (1.4) is called the examinee posterior distribution (conditional on $(\boldsymbol{\Gamma}, \boldsymbol{\Sigma})$).

Mislevy(1984, 1985) formed an EM algorithm, Dempster, Laird, and Rubin(1977), to estimate $(\boldsymbol{\Gamma}, \boldsymbol{\Sigma})$ by considering the $\boldsymbol{\theta}_i$ as missing. If $\boldsymbol{\theta}_i$ were observed, the maximum likelihood estimators would be

$$\hat{\Gamma} = \left[\hat{\gamma}_1 \mid \ldots \mid \hat{\gamma}_p\right] = (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}' \begin{bmatrix} \boldsymbol{\theta}_1' \\ \vdots \\ \boldsymbol{\theta}_n' \end{bmatrix} , \tag{1.5}$$

$$\boldsymbol{S} = \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{\theta}_i - \hat{\boldsymbol{\Gamma}}' \boldsymbol{x}_i\right) \left(\boldsymbol{\theta}_i - \hat{\boldsymbol{\Gamma}}' \boldsymbol{x}_i\right)' . \tag{1.6}$$

When the sufficient statistics for the complete data estimators in (1.5) and (1.6) are replaced by their expected values conditional on the data and provisional parameter values, $(\boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t)$, straightforward calculations similar to those in Little and Rubin(1987) show that

the updated parameter estimates are

$$\boldsymbol{\Gamma}_{t+1} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\begin{bmatrix} \overline{\boldsymbol{\theta}}_1' \\ \vdots \\ \overline{\boldsymbol{\theta}}_n' \end{bmatrix}, \tag{1.7}$$

$$\boldsymbol{\Sigma}_{t+1} = \frac{1}{n}\left[\sum_{i=1}^{n} \mathrm{var}\left(\boldsymbol{\theta}_i \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t\right) + \sum_{i=1}^{n}\left(\overline{\boldsymbol{\theta}}_i - \boldsymbol{\Gamma}_{t+1}'\boldsymbol{x}_i\right)\left(\overline{\boldsymbol{\theta}}_i - \boldsymbol{\Gamma}_{t+1}'\boldsymbol{x}_i\right)'\right]. \tag{1.8}$$

where $\overline{\boldsymbol{\theta}}_i = \left(\overline{\theta}_{i1}, \ldots, \overline{\theta}_{ip}\right)' = \mathrm{E}\left(\boldsymbol{\theta}_i \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t\right)$ is the examinee posterior mean. This process is repeated until convergence.

Equation (1.8) decomposes $\boldsymbol{\Sigma}_{t+1}$ into a component representing the remaining uncertainty in $\boldsymbol{\theta}_i$ after observing $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and the variation of the estimates of $\boldsymbol{\theta}_i$ about the regression of $\boldsymbol{\theta}$ on $\boldsymbol{X}$. When the examination items, $\boldsymbol{y}_i$, determine $\boldsymbol{\theta}_i$ so that $\overline{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i$, there is no remaining uncertainty about $\boldsymbol{\theta}_i$ and the EM algorithm becomes a standard multivariate multiple regression. In the other limiting case when $\boldsymbol{y}_i$ contains no information about $\boldsymbol{\theta}_i$, then $\overline{\boldsymbol{\theta}}_i = \boldsymbol{\Gamma}_t'\boldsymbol{x}_i$, $\boldsymbol{\Gamma}_{t+1} = \boldsymbol{\Gamma}_t$, and $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t$.

An outline of the paper follows. Section 2 presents an example using NAEP data with emphasis on the difference in the results obtained using different numerical approximations for the expectations in the E-step of the EM algorithm. Section 3 describes a method for computing the examinee posterior moments required for the E-step using a common first order asymptotic correction for the examinee posterior means, and a specialized second order asymptotic correction derived for the examinee posterior variance-covariance matrix. Section 4 describes the two alternative approaches illustrated in Section 2 for computing the examinee posterior moments. Section 5 summarizes the relative computing speed of the different methods. Section 6 contains a brief overiveiw of the MGROUP program. Sections 7, 8, and 9 give a detailed description of the different E-step algorithms.

2 Example

Table 1 presents results using data from the main NAEP sample of 8790 students for the 1990 assessment of mathematics for students in the fourth grade or nine years of age. Results are presented for the *Numbers and Operations* scale, in which examinees are given between

19 and 26 examination items, and the *Measurement* scale, in which examinees are given between six and eleven examination items. An example with only two scales is presented because it is possible to apply nearly exact numerical quadrature in this setting. The example uses a subset of the complete set of NAEP conditioning "$\boldsymbol{X}$" variables. There are eleven dichotomous variables representing additive factors for sex, race, modal age-grade status, and teacher rating of class ability. A description of the covariate labels in Table 1 is given in Table 2.

The sample size for typical NAEP applications of the model in Section 1 in the various surveys range from 1000 to 10,000 examinees, with one to six scales. The EM algorithm often requires 100 or more iterations to reach convergence, thus requiring the evaluation of as many as (10,000 examinees) * (100 iterations) * (6 means + 21 variances and covariances) = 27 million six dimensional integrals. As a consequence, methods for evaluating these integrals must be very efficient.

Table 1 displays the converged estimates of $(\boldsymbol{\Gamma}, \boldsymbol{\Sigma})$ from the EM algorithm based on three different methods of evaluating the examinee posterior moments. The estimates in the first two columns of Table 1 are based on a simple numerical quadrature method for the E-step described in Sections 4 and 7. By using additional quadrature points, and simulation studies, the numerical accuracy of this method has been confirmed. However, the quadrature approach is very slow in applications with two scales (four to eight hours for a typical application on a Sun SPARC Station IPX), and computationally unfeasible when there are more than two scales.

The second set of estimates displayed in the middle columns of Table 1 are based on an *Integrated* normal approximation motivated by the asymptotic normality of the examinee likelihood function, $l(\boldsymbol{\theta})$, as the number of examination items increases. This approach, which is described in Sections 4 and 8, produces a very fast algorithm useful for checking purposes and obtaining starting values. For the number of examination items commonly administered, however, the accuracy of this algorithm is inadequate. Discrepancies between the nearly correct estimates obtained using numerical quadrature and the estimates ob-

Table 1: Parameter estimates for the mathematics example

	Numerical Quadrature		Integrated Normal Approximation		Higher Order Asymptotic Corrections	
	Num&Op	Measurement	Num&Op	Measurement	Num&Op	Measurement
Intercept	-0.569	-0.357	-0.561 (0.03)	-0.362 (0.04)	-0.568	-0.354
Female	-0.026	-0.114	-0.021 (0.02)	-0.138 (0.02)	-0.026	-0.115
Afric	-0.512	-0.643	-0.552 (0.02)	-0.828 (0.03)	-0.516	-0.651
Hisp	-0.407	-0.442	-0.441 (0.02)	-0.566 (0.03)	-0.410	-0.448
Asian	0.075	0.041	0.070 (0.05)	0.038 (0.06)	0.074	0.042
EMALMG	-0.577	-0.547	-0.614 (0.03)	-0.700 (0.04)	-0.582	-0.552
EMAGMG	0.572	0.483	0.557 (0.16)	0.522 (0.21)	0.573	0.484
GMAEMG	-0.193	-0.156	-0.200 (0.02)	-0.208 (0.03)	-0.194	- 0.157
LMAEMG	0.140	-0.356	0.155 (0.15)	-0.338 (0.20)	0.143	-0.358
CLABH	0.441	0.407	0.447 (0.04)	0.467 (0.06)	0.443	0.406
CLABM	0.012	0.014	0.019 (0.03)	0.022 (0.03)	0.012	0.013
CLABL	-0.190	-0.223	-0.203 (0.04)	-0.273 (0.05)	-0.190	-0.224
	0.364	0.344	0.382	0.456	0.368	0.350
Σ	0.344	0.368	0.456	0.563	0.350	0.374

Note: The numbers in parentheses are standard errors (SE). These SE are also used as approximate SE for the other computational methods.

Table 2: Description of regressor variables

Regressor	Description
Intercept	
Female	One if examinee is female
Afric	One if examinee is African-American
Hisp	One if examinee is Hispanic
EMALMG	Equal to modal age, less than modal grade
EMAGMG	Equal to modal age, greater than modal grade
GMAEMG	Greater than modal age, equal modal grade
LMAEMG	Less than modal age, equal modal grade
CLABH	Teacher rates class ability high
CLABM	Teacher rates class ability medium
CLABL	Teacher rates class ability low

tained using the normal approximation in Table 1 are large in substantive terms and when compared to approximate standard errors of the estimates. These approximation errors do not decrease as the number of examinees in the sample increase. The approximation errors are larger for the *measurement* scale because it is estimated from fewer examination items per examinee. The estimate of the residual variance for this scale is especially poor. The magnitude of these approximation errors are typical of applications with only a few examination items per examinee. The errors in the parameter estimates are a consequence of errors in the approximation of both the examinee posterior means and variances in equations (1.7) and (1.8). The more common *modal* normal approximation to the examinee posterior distribution, described in Section 3, produces worse approximations than the Integrated normal approximation so results for it are not presented in Table 1.

The estimates presented in the final columns of Table 1 are based on the E-step examinee posterior moments computed using Laplace expansions, (Kass and Steffey, 1989, Mosteller and Wallace, 1964, and de Bruijn, 1981). The calculation of first order corrections for the $\overline{\boldsymbol{\theta}}_t$, and the calculation of second order corrections for var $(\boldsymbol{\theta}_t \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t)$ are relatively simple and very efficient for the special structure of the examinee posterior distributions given in (1.1) and (1.4). The results in Table 1 are indicative of the good accuracy obtained from this approach even in difficult settings where examinees receive between five and ten items per scale. Details of this approach to evaluating the E-step integrals are given in Sections 3 and 9.

The asymptotic corrections achieved the marked improvement by producing much more accurate approximations of the integrals computed for each examinee. To demonstrate this fact, the posterior means and variances were computed for each examinee using the three different methods with the same values of $\boldsymbol{\Gamma}_t$ and $\boldsymbol{\Sigma}_t$ equal to the converged value of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$ based on the numerical quadrature method. The root mean squares of the differences between the means, variances, and the covariance computed for each examinee using the asymptotic corrections and the numerical quadrature range from 90 to 95 percent smaller than the corresponding differences between the Integrated normal approximation

and the numerical quadrature. The performance of the Integrated normal approximation was especially poor for the variance of the shorter *measurement* scale. Figure 1 displays a plot of the posterior variances of 500 randomly selected examinees computed using the Integrated normal approximation versus the posterior variances computed using numerical quadrature. The Integrated normal approximation over-estimates the larger variances. This approximation error inflates the estimates of $\boldsymbol{\Sigma}$ and it also produces a tendency for the magnitude of the $\boldsymbol{\Gamma}$ estimates to be too large as the EM algorithm iterates because the subsequent examinee posterior means are not shrunk sufficiently. Figure 2 shows that the asymptotic corrections dramatically reduce the approximation errors in the examinee posterior variances.

3 Higher order asymptotic corrections for the examinee posterior moments

The EM algorithm in (1.7) and (1.8) requires the computation of the examinee posterior mean $\overline{\boldsymbol{\theta}} = \mathrm{E}\left(\boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t\right)$ and $\mathrm{var}\left(\boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t\right)$. A computationally efficient approach to approximating these moments uses asymptotic approximations as the number of examination items increases. Denote the negative log posterior distribution for an examinee with covariates, $\boldsymbol{x}$, and examination item responses, $\boldsymbol{y}$, by $h\left(\boldsymbol{\theta}\right) = -\log[f\left(\boldsymbol{y} \mid \boldsymbol{\theta}\right) N(\boldsymbol{\theta}; \boldsymbol{\Gamma}_t'\boldsymbol{x}, \boldsymbol{\Sigma}_t)]$, and the total number of items by K. For a sequence of examination items increasing proportionately for each scale, the common *modal* (Bayesian) asymptotic approximation for the posterior mean and the variance-covariance matrix based on Laplace's method, Kass and Steffey(1989), has the form

$$\overline{\theta}_j = \hat{\theta}_j + O\left(1/K\right) , \tag{3.1}$$

$$\mathrm{cov}\left(\theta_j, \theta_k \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_{t+1}, \boldsymbol{\Sigma}_{t+1}\right) = G_{jk} + O\left(1/K^2\right) , \tag{3.2}$$

where $\hat{\theta}_j$ is the posterior mode of θ_j and $G_{jk} = \left(\frac{\partial h}{\partial \theta_r \partial \theta_s}\Big|_{\hat{\theta}}\right)^{-1}_{jk}$. The $\hat{\theta}_j$ are found by Newton-Raphson maximization requiring between two and three iterations. A rigorous statement and derivation of (3.1) and (3.2) and the asymptotic claims that follow are more difficult

for educational measurement models because examinees do not have identically distributed responses to differing examination items. Modifications of the usual asymptotic derivations that account for the varying characteristics of the examination items and the estimation of item specific parameters, like those in Haberman(1977), are not developed here.

Higher order asymptotic corrections are obtained by computing the leading omitted terms in (3.1) and (3.2). For the special structure of h implied by (1.1) and (1.4), the computation of the correction terms is greatly simplified because the examinee likelihood function in (1.1) factors into distinct functions of $(\theta_1, \ldots, \theta_p)$ so that

$$\frac{\partial \log [l(\boldsymbol{\theta})]}{\partial \theta_j \partial \theta_k} = 0 \qquad j \neq k \,,$$

and likewise for all higher order mixed derivatives. In addition, the normal prior distribution for $\boldsymbol{\theta}$ implies that all third and higher derivatives of the log prior distribution are zero, so that the only non-zero derivatives of third and higher order of h are the pure partial derivatives denoted by $\hat{h}_j^{(r)}$, $j = 1, \ldots p$, and $r = 3, 4, \ldots$, where the hat indicates that the derivatives of h are evaluated at $\hat{\boldsymbol{\theta}}$.

Using the special form of the derivatives of h, the first order correction to the mean in (3.1) is given by

$$\overline{\theta}_j \approx \hat{\theta}_j - \frac{1}{2} \sum_{r=1}^{p} G_{jr} G_{rr} \hat{h}_r^{(3)} \,, \tag{3.3}$$

which is a specialization of the first order correction formula given by several authors and summarized in Kass and Steffey(1989). The higher derivatives of the *log* of the examinee likelihood function for the item response models in (1.2) and (1.3) can be computed efficiently, and these derivatives can be reused for the moments of each scale. Because only pure partial derivatives are required in (3.3), a direct approach to computing asymptotic corrections is more efficient in this setting than alternative approaches that avoid the calculation of higher derivatives such as Tierney, Kass, and Kadane(1989).

To improve the usual asymptotic variance-covariance matrix approximations in (3.2), the second order correction term was derived using the symbolic algebra program *Mathematica*. The Laplace expansion was applied to the integrals represented in *standard form*. Tierney,

Kass, and Kadane(1989) yielding the approximation

$$\begin{aligned}
&\operatorname{cov}\left(\theta_j, \theta_k \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_{t+1}, \boldsymbol{\Sigma}_{t+1}\right) \\
\approx\ & G_{jk} - \frac{1}{2}\sum_{r=1}^{p}\left(G_{jr}G_{kr}G_{rr}\right)\hat{h}_r^{(4)} + \frac{1}{2}\sum_{r=1}^{p}\sum_{s=1}^{r}\left\{\left[1 - \frac{1}{2}I\left(r=s\right)\right]\hat{h}_r^{(3)}\hat{h}_s^{(3)}G_{rs}\right. \\
&\left.\times\left(G_{js}G_{ks}G_{rr} + G_{js}G_{kr}G_{rs} + G_{jr}G_{ks}G_{rs} + G_{jr}G_{kr}G_{ss}\right)\right\}. \qquad (3.4)
\end{aligned}$$

The derivation of (3.4) for the case with only one scale is presented in Appendix A, and extensions to the multi-scale case are indicated.

When there is only one scale, (3.4) specializes to

$$\operatorname{var}\left(\theta_1 \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_{t+1}, \boldsymbol{\Sigma}_{t+1}\right) \approx G_{11} - \frac{1}{2}G_{11}^3\hat{h}_1^{(4)} + G_{11}^4\left(\hat{h}_1^{(3)}\right)^2. \qquad (3.5)$$

As anticipated, the correction does not depend on the sign of $\hat{h}_1^{(3)}$, and it increases the variance when the examinee posterior density decays slower than the normal distribution in the neighborhood of $\hat{\theta}$. When applied to the beta, gamma, and t_ν families of distributions, (3.5) produces corrections exactly equal to the corrections based on the expansion of the moment generating function in Tierney, Kass, and Kadane(1989). A simple extension of their Theorem 3 shows that (3.5) and the moment generating function approach produces identical variance formulas in one dimensional problems whenever the expansions are applicable.

4 Alternative methods of computing the examinee posterior moments

Numerical quadrature provides an accurate and computationally feasible approach for evaluating the examinee posterior moments required by the E-step for applications with one or two scales. The quadrature implemented for applications with one scale evaluates the examinee likelihood function on a uniformly spaced grid of n_q points, $\left(l_1\left(q_1\right), q_1\right), \ldots,$ $\left(l_1\left(q_{n_q}\right), q_{n_q}\right)$, and applies the formulas

$$\bar{\theta}_1 \approx \frac{\sum_{i=1}^{n_q} q_i l_1\left(q_i\right) N\left(q_i;\ \boldsymbol{\Gamma}_t'\boldsymbol{x}_1, \boldsymbol{\Sigma}\right)}{\sum_{i=1}^{n_q} l_1\left(q_i\right) N\left(q_i;\ \boldsymbol{\Gamma}_t'\boldsymbol{x}_1, \boldsymbol{\Sigma}\right)} \qquad (4.1)$$

$$\text{var}\left(\theta_1 \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t\right) \approx \frac{\sum_{i=1}^{n_q} \left(q_i - \bar{\theta}_1\right)^2 l_1\left(q_i\right) N\left(q_i;\ \boldsymbol{\Gamma}_t' \boldsymbol{x}_i, \boldsymbol{\Sigma}\right)}{\sum_{i=1}^{n_q} l_1\left(q_i\right) N\left(q_i;\ \boldsymbol{\Gamma}_t' \boldsymbol{x}_i, \boldsymbol{\Sigma}\right)} . \tag{4.2}$$

This approach is described in Beaton(1987). Gaussian quadrature can be used as an alternative which yields similar accuracy with fewer quadrature points, but the Gaussian quadrature is still less efficient because it requires that the examinee likelihood function be re-evaluated at each E-step, in contrast to the uniformly spaced quadrature where the grid of examinee likelihood function evaluations can be performed once before beginning the EM iterations. The generalization to a two dimensional grid of n_q^2 points for two scales is immediate. The numerical quadrature results in Table 1 were computed using a uniformly spaced two dimensional grid of 41^2 points on a square region determined by the values of the item specific parameters. Numerical quadrature routines are computationally unfeasible for applications with more than two scales.

The Integrated normal approximation referred to in Table 1 is much faster than the other computing approximations, but does not produce adequate accuracy for typical applications having scales with less than 20 examination items per examinee. The Integrated normal approach approximates the contribution to the examinee likelihood function associated with each scale by a normal distribution with mean and variance computed using the grid of equally spaced quadrature points,

$$\mu_j = \frac{\sum_{i=1}^{n_q} q_i l_j\left(q_i\right)}{\sum_{i=1}^{n_q} l_j\left(q_i\right)} , \tag{4.3}$$

$$\tau_j = \frac{\sum_{i=1}^{n_q} \left(q_i - \mu_j\right)^2 l_j\left(q_i\right)}{\sum_{i=1}^{n_q} l_j\left(q_i\right)} , \qquad j = 1, \ldots, p. \tag{4.4}$$

Because the examinee likelihood function factors, the examinee likelihood function is approximated by a multivariate normal distribution with a diagonal variance-covariance matrix, $l\left(\boldsymbol{\theta}\right) \approx N\left(\boldsymbol{\mu}, \boldsymbol{D}\right)$ with $\boldsymbol{\mu} = \left(\mu_1, \ldots, \mu_p\right)'$, and $\boldsymbol{D} = \text{diag}\left(\tau_1, \ldots, \tau_p\right)$. The examinee posterior mean and variance-covariance matrix in each E-step are computed using Bayesian normal theory calculations, Box and Tiao(1973),

$$\text{var}\left(\boldsymbol{\theta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t\right) \approx \left(\boldsymbol{\Sigma}_t^{-1} + \boldsymbol{D}^{-1}\right)^{-1} \equiv \boldsymbol{V} . \tag{4.5}$$

$$\overline{\theta} = V\left(\Sigma_i^{-1}\Gamma_i' x + D^{-1}\mu\right) . \tag{4.6}$$

Mislevy, Johnson, and Muraki(1992), and Mislevy(1990) describe this approach. The Integrated normal approximation produces more accurate results than the common modal approximation in (3.1) and (3.2), and the Integrated normal approximation is much faster because the moments for approximating the examinee likelihood function can be evaluated once before the EM iterations begin.

5 Relative speed of the different methods

Asymptotic corrections to the normal approximation have accuracy close to that of extensive product quadrature rules and the corrections are applicable when there are many scales. Timing runs show that the evaluation of the partial derivatives of the examinee log posterior density for all scales requires about the same amount of time as the evaluation of the examinee log posterior density at two distinct values of θ. A Newton-Raphson iteration thus requires the equivalent of two posterior density evaluations, and likewise for the computation of the correction formulas. Three Newton-Raphson iterations yield sufficient accuracy for $\hat{\theta}$ resulting in an algorithm requiring the equivalent of eight posterior density evaluations. The evaluation of the examinee log posterior density grows linearly in the number of examination items (and scales) producing a very efficient algorithm for problems involving multi-dimensional θ_i.

The Integrated normal algorithm is approximately ten times as fast as the numerical quadrature algorithm in two dimensions. The Integrated normal algorithm is approximately five times as fast as the asymptotic correction algorithm regardless of dimension. Applications of the asymptotic correction algorithm typically require one to four hours on a Sun SPARC Station IPX.

6 Overview of the MGROUP program

MGROUP is a FORTRAN77 program with many subroutines that are called by two primary controlling routines called PHASE1 and PHASE2. Mislevy and Sheehan(1987) provides

documentation for earlier versions of the program. The PHASE1 routine calls subroutines that input data and perform calculations that can be done once before beginning the EM iterations. The PHASE2 routine directs the EM iterations, and then executes subroutines to report on the converged values. If requested, PHASE2 also executes subroutines to compute standard errors for Γ and generated imputed values for each examinee. The code for PHASE1 and PHASE2 are given in Appendices B and C. The most important computational steps are briefly outlined in this section; the outlined steps are highlighted in the code for easy reference.

There are currently three versions of the MGROUP program that implement the different methods of approximating the E-step expectations: BGROUP, which implements the numerical quadrature approach in (4.1) and (4.2), NGROUP which implements the Integrated normal approximation in (4.6) and (4.5), and CGROUP which implements the asymptotic correction formulas in (3.3) and (3.4). The different versions of the MGROUP program share most subroutines. Subroutines that are specialized for different versions of MGROUP are indicated by inserting a 'B', 'C', or 'N' in the subroutine name, for example, BPHASE2.f, CPHASE2.f, NPHASE2.f. The function of the specialized subroutines is the same for each version of MGROUP, but the data structures and output are modified to accomadate the differing requirements of the E-step calculations. Subroutines displayed in the appendices are for CGROUP v.2.2, except where explicitly indicated otherwise.

PHASE1

1. Input the item parameters, scale definitions, conditioning variable definitions, and the group definitions (if requested).

2. Input the item response data and the conditioning variables (READDF/READMF). For each examinee, evaluate the equally spaced grid of likelihood values described in Section 4. Use this grid of points to obtain (4.3) and (4.4). Output a work file containing the conditioning variables for use in the E-step. The BGROUP version also

outputs the grid of likelihood values for each examinee in the work file. The examinee likelihood function is normed to have a maximum value of one. The NGROUP version appends only (4.3) and (4.4) to the conditioning variables. The CGROUP version appends (4.3) and (4.4) for use in creating starting values to find the mode in (3.2) for each examinee in the E-step. The item responses are included in the CGROUP work file so that the derivatives of the examinee log likelihood function can be computed. The CGROUP version also includes the maximum value of the (un-normed) examinee log likelihood function for use in norming the approximate log likelihood function for $\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$.

3. Compute $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ for use in the M-step of the EM algorithm.

4. Input starting values for $\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$ or compute them using (4.3) regressed on $\boldsymbol{X}$ using summary statistics accumulated during steps 2 and 3.

PHASE2

1. EM iterations.

   (a) Execute the subroutine ESTEP which computes the examinee posterior mean and variance and accumulates the required summaries of the moments.

   (b) Execute the subroutine MSTEP which is very simple because $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ is already computed.

   (c) Check the covergence criteria. Two criteria are checked: 1) the largest change in $\boldsymbol{\Gamma}$ after dividing by the standard deviations of $\boldsymbol{X}$, 2) the change in the log likelihood function.

2. Produce printed summaries of converged values and write $\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$ to a file if requested.

3. Produce standard errors for $\boldsymbol{\Gamma}$ if requested.

4. Generate multiple imputations if requested. The imputations are produced by the subroutine WRITSF. This subroutine uses the same code as in the ESTEP to compute an approximate posterior mean and variance for each examinee; the subroutine then generates imputations from a normal random variable with the computed mean and variance. Changes to the code of the E-step require corresponding changes in WRITSF.

7 Numerical integration: BGROUP E-step

The code implementing the steps outlined below is given in Appendix D.

1. Read the data from the work file created by READDF/READMF in PHASE1. The data are read one examinee at a time and steps 2-8 are completed for each examinee.

2. Compute the prior mean using Γ_t from the previous iteration and the conditioning variables from the PHASE1 work file.

3. Evaluate the examinee posterior density given in the integrand of (1.4) at each of the uniformly spaced quadrature points created in PHASE1. The density is only determined up to a norming factor at this point.

   (a) Compute the normal prior density for the first scale.

   (b) If there are two scales, then compute the conditional prior distribution for the second scale conditional on the quadrature value of the first scale for each quadrature point in the second scale.

   (c) When there are two scales, multiply 3a) and 3b) to obtain the prior distribution. Multiply the prior by the likelihood function for each scale at the corresponding quadrature points. The likelihood function is obtained from the PHASE1 work file.

4. Accumulate the first and second moments for (4.1) and (4.2). These moments are only determined up to a norming constant.

5. Compute the norming factor by summing the posterior density evaluated at each quadrature point.

6. Accumulate the examinee contribution to the log likelihood function in (1.4) approximating the integral with the uniformly spaced quadrature. Note that $\boldsymbol{\Gamma}_t$ and $\boldsymbol{\Sigma}_t$, which are the estimates from the previous iteration, are used so that the likelihood function evaluations always lag the estimates by one iteration.

7. Center the posterior variance calculations and divide by the norming constant.

8. Accumulate the posterior means and variances for (1.7) and (1.8). Note that $\boldsymbol{\Gamma}_t$ and not $\boldsymbol{\Gamma}_{t+1}$ is used in the centering of (1.8).

8 Integrated normal approximation: NGROUP E-step

The code implementing the steps outlined below is given in Appendix E.

1. Read the data from the work file created by READDF/READMF in PHASE1. The data are read one examinee at a time and steps 2-6 are completed for each examinee.

2. Compute the prior mean using $\boldsymbol{\Gamma}_t$ from the previous iteration and the conditioning variables from the PHASE1 work file.

3. Compute the posterior variance using (4.5).

4. Compute the posterior mean using (4.6).

5. Accumulate the posterior means and variances for (1.7) and (1.8). Note that $\boldsymbol{\Gamma}_t$ and not $\boldsymbol{\Gamma}_{t+1}$ is used in the centering of (1.8).

6. Accumulate the examinee contribution to the log likelihood function in (1.4). The Integrated normal approximation in Section 4 is equivalent to a measurement error model with "observation" $\boldsymbol{\mu}_i$ distributed as $N(\boldsymbol{\theta}_i, \boldsymbol{D}_i)$, and $\boldsymbol{\theta}_i$ distributed as

$N(\Gamma' x_i, \Sigma)$. Thus, $\mu_i \sim N(\Gamma' x_i, \Sigma + D_i)$, and the contribution to the likelihood function in (1.4) with (1.1) approximated by this normal density at (Γ_t, Σ_t) is

$$\frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_t + D_i| - \frac{1}{2}(\mu_i - \Gamma_t' x_i)(\Sigma_t + D_i)^{-1}(\mu_i - \Gamma_t' x_i) \quad . \tag{8.1}$$

Note that $(p/2)\log(2\pi)$ is added to (8.1) to obtain consistency with the BGROUP likelihood calculation which has the maximum of the examinee likelihood function set to one. Also note that Γ_t and Σ_t, which are the estimates from the previous iteration, are used so that the likelihood function evaluations always lag the estimates by one iteration.

9 Asymptotic corrections: CGROUP E-step

The code implementing the steps outlined below is given in Appendix F.

1. Read the data from the work file created by READDF/READMF in PHASE1. The data are read one examinee at a time and steps 2-8 are completed for each examinee.

2. Compute the prior mean using Γ_t from the previous iteration and the conditioning variables from the PHASE1 work file.

3. Complete the NGROUP steps 3 and 4 to produce starting values for finding $\hat{\theta}$.

4. Make a copy of the NGROUP estimates for use if the asymptotic corrections fail. The posterior standard deviations from NGROUP are also used for bounding the amount of change allowed in the Newton-Raphson iterations and the asymptotic corrections. The current value of the step size for scale J stored in `XSTEP(J)` is one posterior standard deviation. The maximum step size can be changed by modifying the value of `XNR1` in the subroutine GETGLO.

5. Perform Newton-Raphson iterations to find $\hat{\theta}$.

   (a) Compute the first and second derivatives of the examinee log likelihood function and the log posterior distribution.

(b) Compute the Cholesky decomposition of the matrix of second derivatives of the log posterior distribution and use it to compute the Newton-Raphson update. The Cholesky subroutine DCHOL returns an appropriately chosen "square root" matrix when the matrix of second derivatives is not positive definite; the algorithm employed is described in Kennedy and Gentleman(1980), p. 442-450, with bounds obtained from the NGROUP posterior variance approximation.

(c) Update the Newton-Raphson approximation to the mode limiting the change in scale J to size `XSTEP(J)`, and check for convergence. The maximum number of iterations allowed is `MAXIT=6` and the current convergence criteria is `XNR2=0.001`; these values are set in the subroutine GETGLO.

6. Compute the asymptotic corrections in Section 3.

(a) Compute G by inverting the second partial derivatives of the log posterior distribution computed at the last Newton-Raphson iteration. Also compute the third and fourth derivatives of the log posterior (which equal to the derivatives of the log likelihood) at the converged value of $\hat{\theta}$.

(b) Compute the asymptotic corrections to $\hat{\theta}$ in (3.3).

(c) If the difference between the corrected approximation to the posterior mean and the NGROUP approximation to the posterior mean exceeds `XNR4*XSTEP`, then set the corrected value equal to the NGROUP approximation plus the appropriately signed value of `XNR4*XSTEP`. The current value of `XNR4` is 1.0 which is set in the subroutine GETGLO.

(d) Determine the bound on the largest change allowed for the posterior variance for each scale. Only the diagonal elements of the variance matrix approximations are checked. A best preliminary approximation to each variance is formed as follows. Each variance approximation in G is divided by the corresponding NGROUP approximation to the variance and if this ratio is greater than `XNR5`, then the NGROUP approximation is used for bounding the asymptotic correc-

tions, and if the ratio is less than XNR5, a weighted average of these two variance approximations is used (where the weights used are based on several examples). The current value of XNR5 is 1.5 which is set in subroutine GETGLO.

(e) Compute the asymptotic corrections to G given in (3.4).

(f) Bound the size of the asymptotic corrections to the variance. If the ratio of the corrected variance to the best preliminary approximation exceeds XNR5, the corrected variance is set to this upper boundary value. If the ratio of the corrected variance to the best preliminary approximation is less than XNR6, then the corrected variance is set to this lower boundary. The current value of XNR6 is 0.25 which is set in subroutine GETGLO.

(g) If any asymptotic correction to the posterior variances fails, then use the correlations from the NGROUP approximation along with the (bounded) corrected variances to form a combined positive definite approximation to the posterior variance.

7. Accumulate the posterior means and variances for (1.7) and (1.8). Note that $\boldsymbol{\Gamma}_t$ and not $\boldsymbol{\Gamma}_{t+1}$ is used in the centering of (1.8).

8. Compute the contribution of each examinee to the log likelihood function.

(a) If $\hat{\boldsymbol{\theta}}$ could not be found using Newton-Raphson iterations, then apply the calculations from NGROUP step 6 for the log likelihood function.

(b) Calculate the approximate likelihood function using the usual normal modal approximation that omits all terms of order $N^{-1/2}$ and smaller,

$$\log\left\{l\left(\hat{\boldsymbol{\theta}}\right)\right\} - \max_{\boldsymbol{\theta}}\left[\log\left\{l\left(\boldsymbol{\theta}\right)\right\}\right] + \frac{1}{2}\left(\log|G| - \log|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\Gamma}_t'\boldsymbol{x}_i\right)' \boldsymbol{\Sigma}_t^{-1}\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\Gamma}_t'\boldsymbol{x}_i\right) . \tag{9.1}$$

The first term in (9.1) is the log of (1.1) evaluated at the mode of the posterior distribution. It is obtained from the final Newton-Raphson iteration. The second

term in (9.1) is the maximum of the log of the likelihood in (1.1). This value is computed in READDF/READMF and placed in the PHASE1 work file. It differs from the first term because $\hat{\boldsymbol{\theta}}$ maximizes the log posterior distribution and not the log likelihood function. Note that $\boldsymbol{\Gamma}_t$ and not $\boldsymbol{\Gamma}_{t+1}$ is used in the centering of (1.8).

A Appendix

This appendix sketches the derivation of the second order correction for the variance in (3.5) when there is only one scale. Generalizations of the derivation to the multi-scale case are discussed briefly. The extensive calculations required for the expansions were completed using the symbolic algebra program *Mathematica*.

Following the derivation of Mosteller and Wallace(1964), p. 153, make the change of variable $u = K^{1/2}\left(\theta_1 - \hat{\theta}_1\right)$, and modify the definition of h by the factor K^{-1} for convenience in tracking the order of the terms, $h \equiv -K^{-1}\log\left[f\left(\boldsymbol{y} \mid \boldsymbol{\theta}\right) N(\boldsymbol{\theta}; \boldsymbol{x}'\boldsymbol{\Gamma}_t, \boldsymbol{\Sigma}_t)\right]$. The Taylor expansion of h about $\hat{\theta}_1$ is

$$Kh\left(\theta_1\right) = Kh\left(\hat{\theta}_1\right) + \frac{G_{11}^{-1}u^2}{2} + \frac{\hat{h}_1^{(3)}u^3}{6K^{1/2}} + \frac{\hat{h}_1^{(4)}u^4}{24K} + \cdots,$$

recalling that G_{11} is the inverse of the second derivative of h evaluated at $\hat{\theta}_1$. The second non-central posterior moment can thus be written as

$$\mathrm{E}\left(\theta_1^2\right) = \frac{\int e^{-Kh(\theta_1)}\theta_1^2 d\theta_1}{\int e^{-Kh(\theta_1)}d\theta_1} =$$

$$\frac{(2\pi G_{11})^{-1/2}\int \exp\left(-\frac{1}{2}G_{11}^{-1}u^2\right)\exp\left(-\frac{\hat{h}_1^{(3)}u^3}{6K^{1/2}} - \frac{\hat{h}_1^{(4)}u^4}{24K} - \cdots\right)\left(\hat{\theta}_1^2 + 2K^{-1/2}\hat{\theta}_1 u + K^{-1}u^2\right)du}{(2\pi G_{11})^{-1/2}\int \exp\left(-\frac{1}{2}G_{11}^{-1}u^2\right)\exp\left(-\frac{\hat{h}_1^{(3)}u^3}{6K^{1/2}} - \frac{\hat{h}_1^{(4)}u^4}{24K} - \cdots\right)du}. \tag{A.1}$$

The integrals in the numerator and denominator of (A.1) can be viewed as the expectations of complicated functions of a normal random variable U with mean 0 and variance G_{11}. Expanding

$$\exp\left(-\frac{\hat{h}_1^{(3)}u^3}{6K^{1/2}} - \frac{\hat{h}_1^{(4)}u^4}{24K} - \cdots\right) \tag{A.2}$$

as a polynomial in u and computing the resulting higher moments of U term wise produces series representations for the numerator and denominator in (A.1). The Taylor series expansion of a ratio is then applied to the series representations for (A.1) and terms smaller than K^{-2} are deleted yielding

$$\mathrm{E}\left(\theta_1^2\right) \approx \quad \hat{\theta}_1^2 + K^{-1}\left(G_{11} - \hat{\theta}_1 G_{11}^2 \hat{h}_1^{(3)}\right) + K^{-2}\left(\frac{1}{12}\right)G_{11}^3\left[15G_{11}\left(\hat{h}_1^{(3)}\right)^2\right.$$

$$-15\hat{\theta}_1 G_{11}^2 \left(\hat{h}_1^{(3)}\right)^3 - 6\hat{h}_1^{(4)} + 16\hat{\theta}_1 G_{11}\hat{h}_1^{(3)}\hat{h}_1^{(4)} - 3\hat{\theta}_1\hat{h}_1^{(3)}\Big]. \tag{A.3}$$

Similar calculations for $\mathrm{E}(\theta_1)$ produce the second order approximation

$$\begin{aligned}\mathrm{E}(\theta_1) \;\approx\; & \hat{\theta}_1 - K^{-1}\left(\frac{1}{2}\right) G_{11}^2 \hat{h}_1^{(3)} + K^{-2}\left(\frac{1}{24}\right) G_{11}^3 \Big[-15 G_{11}^2 \left(\hat{h}_1^{(3)}\right)^3 \\ & +16 G_{11}\hat{h}_1^{(3)}\hat{h}_1^{(4)} - 3\hat{h}_1^{(5)}\Big]. \end{aligned} \tag{A.4}$$

Applying the identity $\mathrm{var}(\theta_1) = \mathrm{E}(\theta_1^2) - (\mathrm{E}(\theta_1))^2$ to (A.3) and (A.4), deleting terms smaller than K^{-2}, and redefining h without K^{-1} produces (3.5). (Note: The refined correction in (A.4) has not been implemented because (3.3) has performed well in practice and (A.4) requires triple summations in the correction formulas in the multi-scale case and the computation of fifth derivatives).

The multi-scale case proceeds similarly with transformed variables $U_1, \ldots, U_p$ that have a multivariate normal distribution with mean zero and variance-covariance matrix with elements G_{jk}. There is an exponential term of the higher powers of u like (A.2) for each u_j in the multi-scale case, but this is still a tremendous simplification of the general second order expansion because there are no mixed (u_j, u_k) terms of order three or higher in the multivariate Taylor expansion of h. The remaining complication in the multi-scale case is that the calculation of the higher univariate normal distribution moments required for the single scale case are replaced by numerous high order multivariate normal distribution moment calculations in the multi-scale case.

B Appendix Subroutine PHASE1

```
C=====================================================================
C SUBROUTINE PHASE1
C ------------------
C
C PURPOSE: SET DEFAULTS AND/OR READ INPUT FILES FOR STARTING
C          ESTIMATION OF GAMMA
C
C=====================================================================
      SUBROUTINE PHASE1(*)
      INCLUDE 'BIGMAT.FOR'
      INCLUDE 'GLOBAL.FOR'
C
```

C PHASE1 1.

```
C
C##NT
C   CHANGE TO NUMERIC CALL
C
      INTEGER*4 JTIME(3)
C
C
      ICYCL = 0
      CALL ALLOC(KINAM,8,NITM)
      CALL ALLOC(KISLO,8,NITM)
      CALL ALLOC(KITHR,8,NITM)
      CALL ALLOC(KIGUE,8,NITM)
      CALL ALLOC(KIUSE,4,NITM)
      CALL ALLOC(KILOC,4,NITM)
      CALL ALLOC(KILNK,4,NITM)
      CALL ALLOC(KIKEY,4,NITM)
      CALL ALLOC(KIOMT,4,NITM)
      CALL ALLOC(KINPR,4,NITM)
C
C     SPECIAL TO CGROUP VERSION OF THE PROGRAM
C
      CALL ALLOC(KOSUB,4,NITM)
C
C     #NT ALLOCATE POLYTOMOUS ITEM PARAMETERS
C         CHANGED TO MEET AL'S NEW CALLING SEQUENCE
C
      CALL ALLOC(KICAT,8,NITM*MAXCAT)
      CALL ALLOC(KIALT,4,NITM)
      CALL ALLOC(KINCA,4,NITM)
      CALL ALLOC(KTNAM,8,NTEST)
      CALL ALLOC(KTLEN,4,NTEST)
C
      CALL READIF(CM(KINAM),BM(KISLO),BM(KITHR),BM(KIGUE),
     $            BM(KICAT),IM(KIUSE),IM(KILOC),IM(KILNK),
     $            IM(KIALT),IM(KINCA),IM(KIKEY),IM(KIOMT),
```

```
     $            IM(KINPR),CM(KSNAM),IM(KSLEN),SM(KOSUB),
     $            CM(KTNAM),IM(KTLEN),*500)
C
      IF (INSCAL .GT. 0) THEN
         IBTEMP = IBASE
         CALL ALLOC(IWRK1,4,NTOTI)
         CALL GETSCA(CM(KSNAM),IM(KILOC),IM(KILNK),IM(IWRK1),*500)
         IBASE = IBTEMP
      END IF
C
C     #NT CHANGED TO NEW PRINT ROUTINE PER A. ROGERS
C
      NLEFT = 0
      CALL PRTSCA(CM(KSNAM),IM(KSLEN),
     $            CM(KINAM),BM(KISLO),BM(KITHR),BM(KIGUE),BM(KICAT),
     $            IM(KIUSE),IM(KILNK),IM(KILOC),IM(KIALT),IM(KINCA),
     $            IM(KIKEY),IM(KIOMT),IM(KINPR))
C
      MGRP = MAX(NGRP,1)
      MCYCLE = MAX(1,NCYCLE+1)
      CALL ALLOC(KCNAM,8,NRANK)
      CALL ALLOC(KCUSE,4,NRANK)
      CALL ALLOC(KCLOC,4,NRANK)
      CALL ALLOC(KCLAB,8,5*NRANK)
      CALL ALLOC(KGNAM,8,MGRP)
      CALL ALLOC(KLNAM,8,MAXLEV*MGRP)
      CALL ALLOC(KGLEV,4,MGRP)
      IF (CFDISP .EQ. 1 .OR. CFDISP .EQ. 3) THEN
         CALL READCF(CM(KCNAM),IM(KCUSE),IM(KCLOC),CM(KCLAB),
     $               CM(KGNAM),CM(KLNAM),IM(KGLEV),*500)
      ELSE
         IF (INCOND .GT. 0) THEN
            CALL GETCON(CM(KCNAM),IM(KCUSE),IM(KCLOC),CM(KCLAB),*500)
         ELSE
            CALL SETCON(CM(KCNAM),IM(KCUSE),IM(KCLOC),CM(KCLAB))
         END IF
         IF (NGRP .GT. 0 .AND. INGRP .GT. 0) THEN
            CALL GETGRP(CM(KGNAM),CM(KLNAM),IM(KGLEV),*500)
         ELSE
            CALL SETGRP(CM(KGNAM),CM(KLNAM),IM(KGLEV))
         END IF
      END IF
C
C PHASE1  1. END
C
      CALL ALLOC(KQPT,8,NQPT)
      CALL ALLOC(KTDT,8,NRANK*NRANK)
      CALL ALLOC(KCOV,8,NSCALE*NSCALE)
      CALL ALLOC(KTDM,8,NRANK*NSCALE)
      CALL ALLOC(KPMN,8,NSCALE)
```

```
      CALL ALLOC(KPVAR,8,NSCALE)
      CALL ALLOC(KQMN,8,NSCALE)
      CALL ALLOC(KQVAR,8,NSCALE)
      CALL ALLOC(KTMU,8,NRANK)
      CALL ALLOC(KTSD,8,NRANK)
      CALL ALLOC(KLK,8,NQPT*NSCALE)
      CALL ALLOC(KGUN,8,TOTLEV)
      CALL ALLOC(KGWN,8,TOTLEV)
      CALL ALLOC(KGMU,8,NSCALE*TOTLEV*MCYCLE)
      CALL ALLOC(KGSD,8,NSCALE*TOTLEV*MCYCLE)
      CALL ALLOC(KTVEC,4,NRANK)
      CALL ALLOC(KGVEC,2,NTOTG)
      CALL ALLOC(KILO,4,NSCALE)
      CALL ALLOC(KIHI,4,NSCALE)
C
C     ADDED BY NT 9/29/91
C     VARIABLES FOR THE STDERR ROUTINE THAT ARE ACCUMULATED
C     IN READDF AND READMF
C
C
      CALL ALLOC(KTAU,8,NSCALE)
      CALL ALLOC(KTT,8,NRANK*NRANK)
C
C
C     VARIABLE ADDED TO ALLOW DIRECT COMPUTATION OF
C     DERIVATIVES OF THE LIKELIHOOD FUNCTION.
C
      CALL ALLOC(KXITEM,4,NITM)
C
      IBTEMP = IBASE
C
      CALL ALLOC(KCOR,8,NRANK*NRANK)
C
C     ## CHANGED 5/4/92 TO CONFORM TO NEW POLYTOMOUS CATEGORY USAGE
C
      CALL ALLOC(KLOGP,8,NQPT*NITM*(MAXCAT+2))
      CALL ALLOC(KLKI,8,NQPT)
      CALL ALLOC(KQLAB,8,NQPT)
      CALL ALLOC(KXVEC,2,NITM)
      CALL ALLOC(KNTRY,4,NSCALE)
C
C     ## ADDED 10/1/92 FOR ITEM RESPONSE FREQUENCIES
C
C
C  PHASE1  2.
C
      CALL ALLOC(KFREQ,8,NITM*(MAXCAT+3))
      CALL ALLOC(KPCT,8,NITM*(MAXCAT+1))
C
      CALL SETQPT(BM(KQPT ),BM(KISLO),BM(KITHR),BM(KIGUE),BM(KICAT),
```

```
     $            IM(KIUSE),IM(KIALT),IM(KINCA),BM(KLOGP),BM(KLKI ))
C     IF (NDIAG .GE. 3) THEN
C        CALL PRTMAT(BM(KLOGP),NQPT,CM(KQLAB),NQPT,CM(KINAM),NITM,
C    $               'TABLE OF LOG(CORRECT) PROBABILITIES BY ITEM')
C        CALL PRTMAT(BM(KLOGQ),NQPT,CM(KQLAB),NQPT,CM(KINAM),NITM,
C    $               'TABLE OF LOG(INCORRECT) PROBABILITIES BY ITEM')
C        CALL PRTMAT(BM(KLKI ),NQPT,CM(KQLAB),NQPT,CM(KSNAM),1,
C    $               'INITIAL LIKELIHOOD ESTIMATES')
C     END IF
C
      IUNIT = 1
C
      IF (DFDISP .EQ. 1) THEN
         CALL GETINP
         CALL ALLOC(KXC,4,NTOTC)
         CALL ALLOC(KXI,4,NTOTI)
         CALL READDF(BM(KQPT ),BM(KTDT ),BM(KCOV ),BM(KTDM ),BM(KPMN ),
     $               BM(KPVAR),BM(KQMN ),BM(KQVAR),SM(KTVEC),BM(KLK  ),
     $               IM(KILO ),IM(KIHI ),BM(KLOGP),BM(KLKI ),
     $               HM(KXVEC),SM(KXC  ),IM(KXI  ),IM(KNTRY),
     $               IM(KCUSE),IM(KCLOC),IM(KIUSE),IM(KILOC),IM(KILNK),
     $               IM(KINCA),IM(KIKEY),IM(KIOMT),IM(KINPR),CM(KSNAM),
     $               HM(KGVEC),IM(KGLEV),BM(KGUN ),BM(KGWN ),BM(KGMU ),
     $               BM(KGSD ),BM(KFREQ),BM(KTAU ),BM(KTT  ),SM(KXITEM),
     $               SM(KOSUB),*500)
      ELSE IF (MFDISP .EQ. 1) THEN
         CALL READMF(BM(KQPT ),BM(KTDT ),BM(KCOV ),BM(KTDM ),BM(KPMN ),
     $               BM(KPVAR),BM(KQMN ),BM(KQVAR),SM(KTVEC),BM(KLK  ),
     $               IM(KILO ),IM(KIHI ),BM(KLOGP),BM(KLKI ),
     $               HM(KXVEC),IM(KNTRY),IM(KCUSE),IM(KILNK),IM(KINCA),
     $               CM(KSNAM),HM(KGVEC),IM(KGLEV),BM(KGUN ),BM(KGWN ),
     $               BM(KGMU ),BM(KGSD ),BM(KFREQ),
     $               BM(KTAU ),BM(KTT  ),SM(KXITEM),SM(KOSUB),*500)
      END IF
C
C  PHASE1  2.  END
C
C
C     PRINT ITEM FREQUENCY STATISTICS        ADDED 10/1/92
C
      CALL STATS(CM(KTNAM),CM(KINAM),IM(KILNK),IM(KIUSE),IM(KILOC),
     $           IM(KINCA),BM(KFREQ),BM(KPCT),*500)
C
C
C  PHASE1  3.
C
      IBASE = IBTEMP
C
```

```
C     KTT IS THE UNWEIGHTED DESIGN MATRIX USED FOR STANDARD ERROR
C         AND IMPUTATION OF GAMMA
C
      IF (NRANK .GT. 1) THEN
         CALL PRTDES(BM(KTDT ),CM(KCNAM),BM(KCOR ),BM(KTMU ),BM(KTSD ))
         CALL INVDES(BM(KTDT ),BM(KCOR ),IM(KCUSE),*500)
         CALL PRTCON(CM(KCNAM),IM(KCUSE),IM(KCLOC),BM(KTMU ),BM(KTSD ),
     $               CM(KCLAB))
C
C     ADDED BY NT 9/29/91
C
         CALL INVDES(BM(KTT ),BM(KCOR ),IM(KCUSE),*500)

      ELSE
         BM(KTDT) = 1D0/BM(KTDT)
C
C     ADDED BY NT 9/29/91
C
         BM(KTT)=1D0/BM(KTT)
      END IF
C
C  PHASE1  3.  END
C
C
      IF (NGRP .GT. 1) THEN
         CALL PRTGRP(CM(KGNAM),CM(KLNAM),IM(KGLEV),CM(KSNAM),
     $               BM(KGUN ),BM(KGWN ),BM(KGMU ),BM(KGSD ))
      END IF
C
      IF (CFDISP .EQ. 2 .OR. CFDISP .EQ. 3)
     $   CALL WRITCF(CM(KCNAM),IM(KCUSE),IM(KCLOC),CM(KCLAB),
     $               CM(KGNAM),CM(KLNAM),IM(KGLEV),*500)
C
C
C  PHASE1  4.
C
      CALL ALLOC(KGAM,8,NRANK*NSCALE)
      CALL ALLOC(KSIG,8,NSCALE*NSCALE)
      CALL ALLOC(KFIXV,8,NSCALE)
      IF (GFDISP .EQ. 1 .OR. GFDISP .EQ. 3) THEN
         CALL READGF(BM( KGAM),BM( KSIG),BM(KFIXV),*500)
      ELSE
         CALL SETGAM(BM( KGAM),BM( KSIG),BM( KTDT),BM( KTDM),
     $               BM( KCOV),BM(KQVAR),BM(KFIXV))
      END IF
C
C
C  PHASE1  4.  END
```

```
C
C
      NLEFT = 0
      CALL PRTGAM(BM( KGAM),BM( KSIG),BM( KCOV),BM(KPVAR),
     $            CM(KCNAM),CM(KSNAM))
C
C##NT
C     CALL GETTIM(IHR,IMIN,ISEC,IHUN)
      CALL ITIME(JTIME)
      IBHIGH = 4*(IBMAX-1)
      KALLOC = 4*IALLOC
      WRITE(6,9200) JTIME(1),JTIME(2),JTIME(3),IBHIGH,KALLOC
C     WRITE(6,9200) TCHAR,IBHIGH,KALLOC

      IBMAX = IBASE
      RETURN
C
 500  WRITE(6,9000)
      RETURN 1
 9000 FORMAT(' ***   EXECUTION TERMINATED IN SUBROUTINE PHASE1  ***')
 9200 FORMAT('0PHASE 1 COMPLETED AT ',I2.2,':',I2.2,':',I2.2,
     $       '   USING ',I8,' OF ',I8,' BYTES OF WORKSPACE')
C9200 FORMAT('0PHASE 1 COMPLETED AT ',A8,
C    $       '   USING ',I8,' OF ',I8,' BYTES OF WORKSPACE')

C#NT THRU HERE

      END
```

C Appendix Subroutine PHASE2

```
C
C     VERSION OF PHASE2.F CREATED FOR CORRECTED ESTEP
C
C     1/9/92   NEAL THOMAS
C
C
C
C
C
      SUBROUTINE PHASE2(*)
      IMPLICIT REAL*8 (A-H,O-Z)
      INCLUDE 'BIGMAT.FOR'
      INCLUDE 'GLOBAL.FOR'
C##NT
C   CHANGE TO NUMERIC CALL
C
C     INTEGER*4 IHR,IMIN,ISEC,IHUN
      INTEGER*4 JTIME(3)
C
      CHARACTER*30    HEDGAM,HEDSIG
      CHARACTER*80    HEDCHG
C
C     ##NT VARIABLES FOR REPORTING ON THE PERFORMANCE OF THE
C          CORRECTION TO THE NORMAL APPROXIMATION TO THE ESTEP.
C          PASSED TO PHASE2 SO THAT THEY CAN BE REPORTED AT THE
C          END OF THE PRINTING
C
      INTEGER ISUBJ,IFAIL,INOTPD,IFCORM,IFCORV
      REAL*8 AVEIT
C
      DATA HEDGAM    /'ACCELERATORS FOR GAMMA        '/
      DATA HEDSIG    /'ACCELERATORS FOR SIGMA        '/
      DATA HEDCHG
     $ /'CHANGES IN STANDARDIZED GAMMA FROM PREVIOUS CYCLES FOR SCALE:'/
C
      SDIM = DFLOAT(NSCALE)
      GDIM = 0D0
      DO 10 I=1,NRANK
         IF (IM(KCUSE+I-1) .EQ. 1) GDIM = GDIM+SDIM
 10   CONTINUE
      IBTEMP = IBASE
      MCYCL = MAX(1,NCYCLE+1)
      CALL ALLOC(MGAM,8,NRANK*NSCALE*MAX(MCYCL,NSIMP))
      CALL ALLOC(MSIG,8,NSCALE*NSCALE*MCYCL)
      CALL ALLOC(KGACC,8,NRANK*NSCALE)
      CALL ALLOC(KSACC,8,NSCALE)
      CALL ALLOC(KCYCL,8,MCYCL)
      CALL SETACC(BM(KGAM),BM(MGAM),BM(KGACC),
     $            BM(KSIG),BM(MSIG),BM(KSACC),
```

```
     $            CM(KCYCL))
C
C     1 LKTOT FOR EACH CYLCLE, CHANGED BY NEAL THOMAS
C
      CALL ALLOC(KLKT,8,MCYCL)
      CALL ALLOC(KSVAR,8,NSCALE)
      CALL ALLOC(KCOVI,8,NSCALE*NSCALE)
      CALL ALLOC(KCOR,8,NSCALE*NSCALE)
      CALL ALLOC(KINDEX,4,NSCALE)
      CALL ALLOC(KID,4,NSCALE)
C
      CALL ALLOC(KD,8,NSCALE)
C
      CALL ALLOC(KGAMCH,8,NRANK*NSCALE*NLAG)
      CALL ALLOC(KBIGG,8,NLAG)
      CALL ALLOC(KSUMG,8,NLAG)
C
C     ##NT
C     ALLOCATE SPACE FOR DERIVATIVES OF THE LOGLIKELIHOOD AND POSTERIOR
C
      CALL ALLOC(KDERIV,8,4*NSCALE)
      CALL ALLOC(KGSP,8,NSCALE*NSCALE)
      CALL ALLOC(KBACK,8,NSCALE)
      CALL ALLOC(KCB,8,NSCALE*NSCALE)
      CALL ALLOC(KMODE,8,NSCALE)
C
C     ##ADDITIONAL WORK SPACE
C
      CALL ALLOC(KXMI,8,NSCALE)
      CALL ALLOC(KXSTEP,8,NSCALE)
      CALL ALLOC(KSINV,8,NSCALE*NSCALE)
      CALL ALLOC(KWMEAN,8,NSCALE)
C
C     ##NT   WORK SPACE FOR LABEL FOR PRTSUM
C
      CALL ALLOC(KRLAB,8,MCYCL)
C
C
C=====================================================================
C     NOW BEGIN THE E-M CYCLE
C=====================================================================
      IRS = NRANK*NSCALE
      ISS = NSCALE*NSCALE
      JGAM = MGAM
      JGAM1 = JGAM-IRS
      JGAM2 = JGAM1-IRS
      JSIG = MSIG
      JSIG1 = JSIG-ISS
      JSIG2 = JSIG1-ISS
C
C     CHANGED BY NEAL THOMAS SO THAT ALWAYS ONE LKTOT PER ITERATION
```

```
C
      JLKT  = KLKT+1
      JGMU  = KGMU+NSCALE+TOTLEV
      JCYCL = KCYCL+MCYCL-1
C
C     ##NT ASSIGN FLAG INDICATING THE RUN IS A RESTART
C
      IRST=0
      IF( (GFDISP.EQ.1).OR.(GFDISP.EQ.3) )IRST=1
C
C     ##NT
C     LIKELIHOOD CALCULATIONS LAG PARAMETER ESTIMATES BY ONE ITERATION
C     SO THAT THE FIRST ENTRY IN THE LIKELIHOOD SUMMARY IS NOT
C     ASSIGNED UNLESS THE RUN IS A RESTART
C
      IF(IRST.EQ.1)THEN
         BM(KLKT)=XLF
         XLNORM=XLI
      ELSE
         XLNORM=0
      END IF
C
C
      IF (NCYCLE .EQ. 0) GO TO 300
C
C
```

C PHASE2 1.

```
C
 100  ICYCL = ICYCL+1
      JCYCL = JCYCL-1
C
```

C PHASE2 1(A)

```
C
      CALL ESTEP(BM(KCOV   ),BM(KTDM  ),BM(KPMN  ),
     $           BM(KXMI   ),BM(KQMN  ),BM(KQVAR ),SM(KTVEC ),
     $           BM(JLKT   ),
     $           BM(JGAM   ),BM(JSIG  ),BM(KXSTEP),
     $           BM(KCOVI  ),BM(KSINV ),
     $           BM(KWMEAN),BM(KD    ),HM(KGVEC ),
     $           BM(JGMU   ),BM(KGWN  ),BM(KISLO ),BM(KITHR  ),
     $           BM(KIGUE  ),BM(KICAT ),IM(KILNK ),IM(KINCA  ),
     $           SM(KXITEM),BM(KDERIV),
     $           BM(KGSP   ),BM(KBACK ),BM(KCB   ),BM(KMODE  ),
     $           ISUBJ,IFAIL,INOTPD,IFCORM,IFCORV,AVEIT)
C
```

C PHASE2 1(A) END

```
C
C
C                                        UPDATE POINTERS
      JGAM2 = JGAM1
      JGAM1 = JGAM
      JGAM  = JGAM+IRS
      JSIG2 = JSIG1
      JSIG1 = JSIG
      JSIG  = JSIG+ISS
C
C  PHASE2  1(B)
C
      CALL  MSTEP(BM(KTDT  ),BM(KCOV  ),BM(KTDM  ),BM(KFIXV ),
     $            BM(JGAM  ),BM(JSIG  ),BM(KSVAR ))
C                                        ACCELERATE GAMMA
      CALL ACCGAM(BM(JGAM  ),BM(JGAM1 ),BM(JGAM2 ),BM(KGACC ),
     $            BM(KGAMCH),BM(KTSD  ),BM(KBIGG ),BM(KSUMG ))
C                                        ACCELERATE SIGMA
      CALL ACCSIG(BM(JSIG  ),BM(JSIG1 ),BM(JSIG2 ),BM(KSACC ),
     $            BM(KSVAR ),BIGS,SUMS)
C
C  PHASE2  1(B)  END
C
      NLEFT = 0
      CALL PRTGAM(BM(JGAM  ),BM(JSIG  ),BM(KCOR  ),BM(KSVAR ),
     $            CM(KCNAM ),CM(KSNAM ))
C
      IF (NDIAG .GE. 1 .AND. NACC .GT. 0) THEN
         IF(MOD(ICYCL,NACC) .EQ. 0)THEN
            CALL PRTMAT(BM(KGACC),NRANK,CM(KCNAM),NRANK,
     $                  CM(KSNAM),NSCALE,HEDGAM)
            IF (ICYCL .EQ. NACC)
     $         CALL PRTMAT(BM(KSACC),1,'DIAGONAL',1,CM(KSNAM),NSCALE,
     $                     HEDSIG)
         END IF
      END IF
      SUMS = SUMS/SDIM
      KLAG = MIN(ICYCL,NLAG)
      IF (NDIAG .GE. 1) THEN
         JGAMCH = KGAMCH
         DO 180 I=1,NSCALE
            WRITE(HEDCHG(63:70),8000) CM(KSNAM+I-1)
            CALL PRTMAT(BM(JGAMCH),NRANK*NSCALE,CM(KCNAM),NRANK,
     $                  CM(JCYCL),KLAG,HEDCHG)
            JGAMCH = JGAMCH+NRANK
180      CONTINUE
      END IF
```

```
C
C     NORM -2*LOGLIKELIHOOD BY ITS INITIAL VALUE
C
      BM(JLKT)=BM(JLKT)-XLNORM
C
C     ##NT CHANGED TO ALSO OUTPUT -2*LOGLIKELIHOOD AND DIAGNOSTICS
C
      IF (NLEFT .LT. 16) CALL HEDOUT
      WRITE(6,9010) ICYCL,(CM(JCYCL+ILAG-1),ILAG=1,KLAG)
      WRITE(6,9020) (BM(KSUMG+ILAG-1),ILAG=1,KLAG)
      WRITE(6,9030) (BM(KBIGG+ILAG-1),ILAG=1,KLAG)
C
      WRITE(6,9040)BM(JLKT)
      WRITE(6,9050)ISUBJ,IFAIL,INOTPD,IFCORM,IFCORV,AVEIT
C
C     CHANGED BY NEAL THOMAS SO THAT ALWAYS ONE LKTOT PER ITERATION
C
      JLKT = JLKT+1
      JGMU = JGMU+NSCALE*TOTLEV
      BIGG = DABS(BM(KBIGG))
C
C  PHASE2  1(C)
C
C     ##NT  CHECK CONVERGENCE
C           ADDED CRITERIA BASED ON -2*LOG(LIK)
C     ##NT  MODIFIED CONVERGENCE CRITERIA AGAIN 10/22/92
C
C     FIRST ITERATION CONVERGENCE CHECKS HANDLED DIFFERENTLY
C     DEPENDING ON WHETHER THE RUN IS A NEW JOB OR RESTART
C
C     NEW RUN (DON'T CHECK LIKELIHOOD CRITERIA)
C     NOTE THAT XLNORM IS 0 UNTIL SET HERE FOR NEW RUNS
C
      IF( (IRST.EQ.0).AND.(ICYCL.EQ.1) )THEN
         XLNORM=BM(JLKT-1)
         XLI=XLNORM
         BM(JLKT-1)=BM(JLKT-1)-XLNORM
C
C
C     RESTART OR ICYCL GREATER THAN 1
C
      ELSE IF( (DABS(BM(JLKT-2)-BM(JLKT-1)).GT.CRITL) .AND.
     $        (ICYCL .LT. NCYCLE) ) THEN
         GO TO 100
      END IF
C
C
      IF (BIGG .GT. CRIT .AND. ICYCL .LT. NCYCLE) GO TO 100
C
C . PHASE2  1.  END
```

```
C
C
 200  CONTINUE
C
C
C  PHASE2  2.
C
C     ##NT UPDATE FINAL LIKELIHOOD FOR EXTERNAL STORAGE
C
      XLF=BM(JLKT-1)
C
      NCYCLE = ICYCL+1
      IF (GFDISP .EQ. 2 .OR. GFDISP .EQ. 3) THEN
         CALL WRITGF(BM(JGAM  ),BM(JSIG  ))
      END IF
      CALL ALLOC(KFLAG,4,NCYCLE*NRANK)
      CALL PRTSUM(BM(MGAM  ),BM(MSIG  ),BM(KLKT  ),BM(KTSD  ),
     $            CM(KSNAM ),CM(KCNAM ),SM(KFLAG ),IM(KGLEV ),
     $            CM(KGNAM ),CM(KLNAM ),BM(KGMU  ),CM(KRLAB ))
C
C  PHASE2  2.  END
C
C
C     #NT
C        FINAL VALUES FOR GAMMA AND SIGMA COPIED INTO WORK
C        SPACE INITIALLY SET ASIDE FOR THESE VALUES.  THIS ALLOWS
C        THE ARRAYS HOLDING THE HISTORY OF THESE VARIABLES TO BE
C        REUSED AS WORK SPACE BY WRITSF
C
      DO 210 KK=0,NSCALE*NRANK-1
         BM(KGAM+KK) = BM(JGAM+KK)
 210  CONTINUE
      DO 220 KK=0,NSCALE*NSCALE-1
         BM(KSIG+KK) = BM(JSIG+KK)
 220  CONTINUE
C
C
C  PHASE2  3.
C
C
      IF (SE .EQ. 1) THEN
C
C        # NT ON 9/29/91
C
         CALL ALLOC(KGSE,8,NRANK*NSCALE)
C
         CALL STDERR(BM(  KTT ),BM(KGAM  ),BM(  KSIG),BM( KTAU),
     $               BM(  KGSE),CM( KCNAM),CM( KSNAM))
```

```
      END IF
C
C  PHASE2  3.  END
C
 300  CONTINUE
C
C  PHASE2  4.
C
      IF (SFDISP .EQ. 2) THEN
            CALL ALLOC(KSIGE,8,NSCALE*NSCALE)
            CALL ALLOC(KSIGI,8,NSCALE*NSCALE)
            CALL ALLOC(KBETA,8,NSCALE*NRANK)
            CALL ALLOC(KBIMP,8,NSCALE*NRANK)
            CALL ALLOC(KTHETA,8,NSIMP*NSCALE)
C
C           CHANGED BY NT FOR USE IN MODIFIED WRITESF
            MAXRS=MAX(NSCALE,NRANK)
            CALL ALLOC(KZ,8,MAXRS)
C
C           THE VARIABLE TDT IS REUSED AS WORKSPACE FOR
C           WRITSF IN ORDER TO AVOID ALLOCATING ANOTHER
C           NRANK BY NRANK MATRIX.  THE VARIABLE HOLDING
C           THE GAMMA'S ESTIMATED FOR EACH CYCLE, MGAM, IS ALSO
C           REUSED BY WRITSF.
C
C
            CALL WRITSF(BM(KQVAR ),BM(KXMI  ),BM(KTAU  ),
     $                  SM(KTVEC ),BM(MGAM  ),BM(KSIG  ),
     $                  BM(KPMN  ),BM(KCOVI ),BM(KCOR  ),IM(KINDEX),
     $                  IM(KID   ),BM(KZ    ),BM(KTHETA),BM(KGAM  ),
     $                  BM(KSIGE ),BM(KSIGI ),BM(KBETA ),BM(KTT   ),
     $                  BM(KTDT  ),BM(KBIMP ),BM(KSINV ),BM(KQMN  ),
     $                  BM(KWMEAN),BM(KXSTEP),BM(KD    ),BM(KISLO ),
     $                  BM(KITHR ),BM(KIGUE ),BM(KICAT ),IM(KILNK ),
     $                  IM(KINCA ),SM(KXITEM),BM(KDERIV),BM(KGSP  ),
     $                  BM(KBACK ),BM(KCB   ),BM(KMODE ),HM(KGVEC ),
     $                  MAXRS)
      END IF
C
C  PHASE2  4.  END
C
C
C
C##NT
C     CALL GETTIM(IHR,IMIN,ISEC,IHUN)
      CALL ITIME(JTIME)
      IBHIGH = 4*(IBMAX-1)
      KALLOC = 4*IALLOC
```

```
      WRITE(6,9200) JTIME(1),JTIME(2),JTIME(3),IBHIGH,KALLOC
C     WRITE(6,9200) TCHAR,IBHIGH,KALLOC
      IBASE = IBTEMP
      RETURN
 8000 FORMAT(A8)
 9010 FORMAT('-SUMMARY OF STANDARDIZED GAMMA CHANGES AFTER CYCLE',I4/
     $       '0    FROM :',T13,10(4X,A8))
 9020 FORMAT('  AVERAGE :',T13,10F12.5)
 9030 FORMAT('  LARGEST :',T13,10F12.5)
 9200 FORMAT('0PHASE 2 COMPLETED AT ',I2.2,':',I2.2,':',I2.2,
     $       '   USING ',I8,' OF ',I8,' BYTES OF WORKSPACE')
C9200 FORMAT('0PHASE 2 COMPLETED AT ',A8,
C    $       '   USING ',I8,' OF ',I8,' BYTES OF WORKSPACE')
 9040 FORMAT('0 -2*LOG(LIKELIHOOD)=',F14.5)
 9050 FORMAT('0NUMBER OF SUBJECTS ',T60,I6,/,
     $       ' SUBJECTS WHERE MODE NOT FOUND',T60,I6,/,
     $       ' SUBJECTS WHERE INFORMATION NOT POSITIVE DEFINITE',
     $T60,I6,/,' SUBJECTS WHERE CORRECTION TO MEAN EXCEEDED BOUND',
     $T60,I6,/,' SUBJECTS WHERE CORRECTION TO VARIANCE',
     $         ' EXCEEDED BOUND',T60,I6,/,' AVERAGE NUMBER OF',
     $         ' NEWTON-RAPHSON ITERATIONS',T60,F7.3)

C#NT THRU HERE
      END
```

D Appendix Subroutine BESTEP

```
C
C     NUMERICAL INTEGRATION OF 1 AND 2 DIMENSIONAL THETA
C
C
      SUBROUTINE ESTEP(QPT,COV,TDM,PMEAN,QVAR,XMEANI,T,
     $                 LK,LKTOT,GAMMA,SIGMA,
     $                 COVI,GVEC,GMU,GWN)
      IMPLICIT REAL*8 (A-H,O-Z)
      INCLUDE 'BIGMAT.FOR'
      INCLUDE 'GLOBAL.FOR'
      LOGICAL          QDIAG
      INTEGER*2        GVEC(NTOTG)
      REAL*4           T(NRANK)
      REAL*8           QPT(NQPT),COV(NSCALE,NSCALE),TDM(NRANK,NSCALE),
     $                 PMEAN(NSCALE),QVAR(NSCALE),
     $                 XMEANI(NSCALE),
     $                 LK(NQPT,NSCALE),LKTOT,
     $                 GAMMA(NRANK,NSCALE),SIGMA(NSCALE,NSCALE),
     $                 COVI(NSCALE,NSCALE),
     $                 GMU(NSCALE,TOTLEV),
     $                 GWN(TOTLEV)
C
C      QPT   GRID OF QUADRATURE POINTS CREATED IN PHASE1
C
C      COV,TDM  ACCUMALTES SUMS OF EXAMINEE POSTERIOR VARIANCES AND MEANS
C
C      PMEAN   HOLDS PRIOR MEANS COMPUTED FROM INPUTS OF PHASE1 WORK FILE
C              AND GAMMA AND SIGMA FROM PREVIOUS ITERATION
C
C      QVAR    APPROXIMATE "VAR" OF EXAMINEE LIKELIHOOD FUNCTION
C              INPUT FROM PHASE1 WORK FILE.  USED ONLY IN DIAGNOSTIC
C              REPORTING
C
C      XMEANI  EXAMINEE POSTERIOR MEAN
C
C      T       HOLDS CONDITIONING VARIABLES READ FROM PHASE1 WORK FILE
C
C      LK      EXAMINEE LIKELIHOOD FUNCTION EVALUATED ON A GRID FOR
C              EACH SCALE, INPUT FROM PHASE1 WORK FILE
C
C      LKTOT   -2*LOG LIKELIHOOD FUNCTON
C
C      GAMMA,SIGMA   VALUES OF GAMMA AND SIGMA FROM
C              PREVIOUS ITERATION
C
C     COVI    EXAMINEE POSTERIOR VARIANCE
C
C     GMU,GWN  USED TO ACCUMULATE GROUP SUMMARY STATISTICS IF
C             REQUESTED
```

```
C
C      VARIABLES USED TO CALCULATE POSTERIOR MEANS AND VARIANCES
C
      REAL*8          RHO,LKCON,DET,B,S1,S2,P1,P2,POSTI,DIFF
C
C     LKCON           WIDTH OF QUADRATURE INTERVAL OR AREA OF SQUARE
C                     FOR NORMING LIKELIHOOD INTEGRAL CALCULATIONS
C
C     RHO             FOR THE TWO DIMENSIONAL CASE, RHO IS THE PRIOR
C                     CORRELATION BETWEEN SUBSCALES
C
C     DET             DETERMINANT OF PRIOR VARIANCE SIGMA
C
C     B               REGRESSION COEFFICIENT OF SCALE 2 ON SCALE 1
C                     USED FOR COMPUTING PRIOR DISTRIBUTION IN TWO
C                     STEPS
C
C     S1,S2           CONSTANTS FOR COMPUTING PRIOR THAT
C                     INCLUDE NUMERICAL BOUND CHECKS
C
C
C     P1,P2           PRIOR FOR SCALE 1, CONDITIONAL PRIOR FOR SCALE 2
C                     GIVEN SCALE 1
C
C     POSTI           POSTERIOR EVALUATED AT EACH QUADRATURE POINT
C
C     DIFF            NUMERATOR OF PRIOR
C
C
C     THE VARIABLE LKTOT IS NOW ALWAYS UNIDIMENSIONAL
C
C
C
      CHARACTER*30    SUBJID
      CHARACTER*40    HEDCOV,HEDCTM
C
C
C
      DATA HEDCOV    /'ADJUSTED COVARIANCE MATRIX              '/
      DATA HEDCTM    /'CHOLESKY TRANSFORMATION MATRIX          '/
      IF (NSCALE.GT.2) THEN
         WRITE(6,9000)
 9000    FORMAT('1  THIS ESTEP ONLY WORKS WITH 1 AND'
     $          ' 2 DIMENSIONAL THETA.')
         STOP
         END IF
C
      DSML = 1D-6
      DBIG = 14D1
C
C     CONSTANTS NEEDED FOR POSTERIOR CALCULATIONS
```

```
C
      LKCON=(QPMAX-QPMIN)/DFLOAT(NQPT-1)
      IF (NSCALE.EQ.2) LKCON=LKCON*LKCON
C
      IF (NSCALE.EQ.1)DET=SIGMA(1,1)
      IF (NSCALE.EQ.2) DET=SIGMA(1,1)*SIGMA(2,2)-SIGMA(1,2)*SIGMA(2,1)
C
C     PUT IN NUMERIC BOUND AND CHECK FOR LESS THAN FULL RANK SIGMA
C
      IF(DET.LT.DSML)THEN
         DET=DSML
         CALL HEDOUT
         NLEFT=NLEFT-4
         WRITE(6,7100)ICYCL
 7100    FORMAT('0THE DETERMINANT OF SIGMA IS CLOSE TO ZERO. ',
     $          ' ICYCL=',I3,/,' A NUMERIC BOUNDARY VALUE WAS ',
     $          'INSERTED TO PREVENT PROGRAM TERMINATION.',/,
     $          ' OUTPUT THAT FOLLOWS IS UNRELIABLE.')
       END IF
C
C     CONSTANTS FOR COMPUTING THE PRIOR DISTRIBUTION IN
C     TWO STEPS
C
      S1=MAX( DSML,MIN(DBIG,2D0*SIGMA(1,1)) )
      IF(NSCALE.EQ.2)THEN
         RHO=SIGMA(1,2)/DSQRT(SIGMA(1,1)*SIGMA(2,2))
         B=RHO*DSQRT(SIGMA(2,2)/SIGMA(1,1))
         S2=MAX( DSML,MIN(DBIG,2D0*(1-RHO*RHO)*SIGMA(2,2)) )
      END IF
C
C                                        INITIALIZE LKTOT, COV & TDM
C*VDIR: ASSUME COUNT(2)
         LKTOT = 0D0
      DO 40 ISCAL=1,NSCALE
C*VDIR: ASSUME COUNT(2)
         DO 20 JSCAL=1,NSCALE
            COV(JSCAL,ISCAL) = 0D0
 20      CONTINUE
C*VDIR: ASSUME COUNT(50)
         DO 30 IRANK=1,NRANK
            TDM(IRANK,ISCAL) = 0D0
 30      CONTINUE
 40   CONTINUE
C*VDIR: ASSUME COUNT(10)
      DO 60 K=1,TOTLEV
C*VDIR: ASSUME COUNT(2)
         DO 50 ISCAL=1,NSCALE
            GMU(ISCAL,K) = 0D0
 50      CONTINUE
 60   CONTINUE
      REWIND(IUNIT)
```

```
      ISUBJ = 0
C                                        MAIN LOOP OVER SUBJECTS
C
C
C  BESTEP  1.
C
 100  READ(IUNIT,END=500) SUBJID(1:NIDW),KUSE,WGHT,MLO,MHI,
     $                    (T(N),N=1,NRANK),(QVAR(J),J=1,NSCALE),
     $                    ((LK(IQPT,J),IQPT=MLO,MHI),J=1,NSCALE),
     $                    (GVEC(I),I=1,NTOTG)
      IF (KUSE .EQ. 0) GO TO 100
C
C  BESTEP  1.  END
C

      ISUBJ = ISUBJ+1
      QDIAG = ICYCL .EQ. 1 .AND. ISUBJ .LE. NQC
C*VDIR: ASSUME COUNT(2)
C
C
C  BESTEP  2.
C
      DO 120 ISCAL=1,NSCALE
         PMI = 0D0
C*VDIR: ASSUME COUNT(50)
         DO 110 IRANK=1,NRANK
            PMI = PMI+GAMMA(IRANK,ISCAL)*T(IRANK)
 110     CONTINUE
         PMEAN(ISCAL) = PMI
 120  CONTINUE
C
C  BESTEP  2.  END
C
      IF (QDIAG) THEN
         CALL HEDOUT
         WRITE(6,8000) ISUBJ,ICYCL
         NLEFT = 55
      END IF
C                                        LOOP OVER SCALES
C                                        COMPUTE PRIOR & POSTERIOR DISTN'S
C                                        ESTIMATES OF SUBJECT MEANS
C                                        & VARIANCES
C*VDIR: ASSUME COUNT(2)
C
      TOTAL=0D0
```

```
      XMEANI(1)=0D0
      COVI(1,1)=0D0
      IF(NSCALE.EQ.2) THEN
         XMEANI(2)=0D0
         COVI(2,1)=0D0
         COVI(1,2)=0D0
         COVI(2,2)=0D0
      END IF
C
C     COMPUTE PRIOR FOR SCALE 1
C
C  BESTEP  3.
C
      DO 130 IQPT=MLO,MHI
C
C  BESTEP  3(A)
C
         DIFF = (QPT(IQPT)-PMEAN(1))**2
         DIFF = MAX(DSML,MIN(DBIG,DIFF/S1))
         P1 = DEXP(-DIFF)
C
C  BESTEP  3(A)  END
C
C
         IF(NSCALE.EQ.2) THEN
C
C           THE BIVARIATE PRIOR COMPUTED USING THE FACTORIZATION
C           F(THETA1)*F(THETA2|THETA1)
C
C
            DO 140 JQPT=MLO,MHI
C
C  BESTEP  3(B)  (TWO SCALES)
C
               DIFF = (QPT(JQPT)- (PMEAN(2)+B*(QPT(IQPT)
     $                          -PMEAN(1))) )**2
               DIFF = MAX(DSML,MIN(DBIG,DIFF/S2))
               P2=DEXP(-DIFF)
C
C  BESTEP  3(C) (TWO SCALES)
C
               POSTI=P1*P2*LK(IQPT,1)*LK(JQPT,2)
C
C  BESTEP  4. (TWO SCALES)
```

```
C
C
C               IMPLEMENTING EQUATIONS (4.1) AND (4.2)
C
                XMEANI(1)=XMEANI(1)+QPT(IQPT)*POSTI
                XMEANI(2)=XMEANI(2)+QPT(JQPT)*POSTI
                COVI(1,1)=COVI(1,1)+QPT(IQPT)*QPT(IQPT)*POSTI
                COVI(1,2)=COVI(1,2)+QPT(IQPT)*QPT(JQPT)*POSTI
                COVI(2,1)=COVI(1,2)
                COVI(2,2)=COVI(2,2)+QPT(JQPT)*QPT(JQPT)*POSTI
C
C   BESTEP  5.  (TWO SCALES)
C
                TOTAL=TOTAL+POSTI
 140          CONTINUE
C
C         1 DIMENSION ONLY
C
          ELSE
C
C   BESTEP  3(C) (ONE SCALE)
C
            POSTI=P1*LK(IQPT,1)
C
C   BESTEP  4. (ONE SCALE)
C
            XMEANI(1)=XMEANI(1)+QPT(IQPT)*POSTI
            COVI(1,1)=COVI(1,1)+QPT(IQPT)*QPT(IQPT)*POSTI
C   BESTEP  5.  (ONE SCALE)
            TOTAL=TOTAL+POSTI
          END IF
 130   CONTINUE
C
C   BESTEP  6.
C
C      LIKELIHOOD CALCULATION:  NOTE THAT LKCON AND DET
C      COMPUTED ONCE INITIALLY AND DIFFERENTLY FOR 1 AND 2
C      DIMENSIONAL CASES.  COMPUTES -2* LOG OF LIKELIHOOD
C      FUNCTION IN (1.4)
C
C
       LKTOT = LKTOT+WGHT*( DLOG(TOTAL*LKCON)-0.5*DLOG(DET) )
C
C   BESTEP  7.
```

```
C
C     NORM AND CENTER POSTERIOR MOMENTS
C
      XMEANI(1)=XMEANI(1)/TOTAL
      COVI(1,1)=COVI(1,1)/TOTAL
      COVI(1,1)=COVI(1,1)-XMEANI(1)*XMEANI(1)
      IF(NSCALE.EQ.2)THEN
         XMEANI(2)=XMEANI(2)/TOTAL
         COVI(2,2)=COVI(2,2)/TOTAL
         COVI(2,2)=COVI(2,2)-XMEANI(2)*XMEANI(2)
         COVI(2,1)=COVI(2,1)/TOTAL
         COVI(2,1)=COVI(2,1)-XMEANI(2)*XMEANI(1)
         COVI(1,2)=COVI(2,1)
      END IF
C
C  BESTEP  7.  END
C
C
      IF (QDIAG) THEN
         WRITE(6,8030)
         NLEFT = NLEFT-3
C*VDIR: ASSUME COUNT(2)
         DO 210 ISCAL=1,NSCALE
            QQVAR = 0D0
            IF (QVAR(ISCAL) .GT. 0D0) QQVAR = 1D0/QVAR(ISCAL)
            WRITE(6,8040) ISCAL,PMEAN(ISCAL),SIGMA(ISCAL,ISCAL),QQVAR,
     $                    XMEANI(ISCAL),COVI(ISCAL,ISCAL)
            NLEFT = NLEFT-1
  210    CONTINUE
      END IF
C
      IF (QDIAG) THEN
         CALL PRTMAT(COVI,NSCALE,CM(KSNAM),NSCALE,CM(KSNAM),NSCALE,
     $               HEDCOV)
      END IF
C
C
C  BESTEP  8.
C
C        ACCUMULATE TDM AND COV WITHOUT IMPUTATIONS
C        EQUATIONS (1.7) AND (1.8)
C
C
         DO 340 JSCAL=1,NSCALE
           DO 330 IRANK=1,NRANK
              TDM(IRANK,JSCAL) = TDM(IRANK,JSCAL)+
     $                           WGHT*T(IRANK)*XMEANI(JSCAL)
 330       CONTINUE
```

```
          XMFAC=XMEANI(JSCAL)-PMEAN(JSCAL)
C
           DO 350 ISCAL=JSCAL,NSCALE
              COV(ISCAL,JSCAL) = COV(ISCAL,JSCAL)+
     $                              WGHT*COVI(ISCAL,JSCAL)
              COV(ISCAL,JSCAL) = COV(ISCAL,JSCAL)+
     $            WGHT*(XMEANI(ISCAL)-PMEAN(ISCAL))*XMFAC
              COV(JSCAL,ISCAL)=COV(ISCAL,JSCAL)
 350        CONTINUE
 340     CONTINUE
C
```

C BESTEP 8. END

```
C
C
C
C
C    ### REPLACE GROUP MEAN FROM IMPUTATION ESTIMATE WITH
C        GROUP MEAN FROM ANALYTIC CALCULATION
C
      DO 440 K=1,NTOTG
         ILEV = GVEC(K)
         IF (ILEV .GT. 0) THEN
            DO 430 ISCAL=1,NSCALE
               GMU(ISCAL,ILEV) = GMU(ISCAL,ILEV)+WGHT*XMEANI(ISCAL)
 430        CONTINUE
         END IF
 440  CONTINUE
C
      GO TO 100
 500  CONTINUE
C
C
C     RETURN -2LOGLIK AND NORM WEIGHTS TO ADD TO SAMPLE SIZE
C
C
      LKTOT=-2D0*LKTOT*(DFLOAT(ISUBJ)/WGTSUM)
C
      DO 530 K=1,TOTLEV
         WT = GWN(K)
         IF (WT .NE. 0D0) THEN
            DO 520 ISCAL=1,NSCALE
               GMU(ISCAL,K) = GMU(ISCAL,K)/WT
 520        CONTINUE
         END IF
 530  CONTINUE
C
      RETURN
C
C
 8000 FORMAT(' <DIAG> PRINTOUT FOR SUBJECT NUMBER ',I4,
```

```
     $       '     CYCLE = ',I4)
 8010 FORMAT('OSCALE',I3,'  THETA        LK     PRIOR      POST',
     $       '    N.POST')
 8020 FORMAT(' ',I5,F10.3,4F10.5)
 8030 FORMAT('0      -----PRIOR---------           -----POSTERIOR-----'
     $ /' SCALE       MU      SIGMA   VAR(LK)      MEAN       VAR ')
 8040 FORMAT(' ',I5,5F10.5)
C
C###  NT FIXING ALTERNATE RETURN CALLS TO SYMINV
C
 7000 WRITE(6,7010)
 7010 FORMAT(' CALL TO SYMINV FAILED IN ESTEP')
      STOP
      END
```

E Appendix Subroutine NESTEP

```
C
C     THIS ESTEP IS MODIFIED TO USE A NORMAL APPROXIMATION
C     FOR MULTIVARIATE ESTEP.  THE NORMAL APPROXIMATION IS
C     DESCRIBED IN THE 1988 TECHNICAL REPORT (MGROUP AT THAT
C     TIME DID NOT ACTUALLY IMPLEMENT THE NORMAL APPROXIMATION
C     DESCRIBED THERE).  ADDITIONAL DOCUMENTATION IS PROVIDED IN
C     "THE E-STEP OF THE EM ALGORITHM"
C
      SUBROUTINE ESTEP(COV,TDM,PMEAN,XMEANI,QMEAN,QVAR,T,
     $                 LKTOT,GAMMA,SIGMA,
     $                 COVI,SINV,WMEAN,GVEC,GMU,GWN)
      IMPLICIT REAL*8 (A-H,O-Z)
      INCLUDE 'BIGMAT.FOR'
      INCLUDE 'GLOBAL.FOR'
      LOGICAL          QDIAG
      INTEGER*2        GVEC(NTOTG)
      REAL*4           T(NRANK)
      REAL*8           COV(NSCALE,NSCALE),TDM(NRANK,NSCALE),
     $                 PMEAN(NSCALE),XMEANI(NSCALE),
     $                 QMEAN(NSCALE),QVAR(NSCALE),LKTOT,
     $                 GAMMA(NRANK,NSCALE),SIGMA(NSCALE,NSCALE),
     $                 COVI(NSCALE,NSCALE),SINV(NSCALE,NSCALE),
     $                 WMEAN(NSCALE),GMU(NSCALE,TOTLEV),
     $                 GWN(TOTLEV)
C
C                     VARIABLES FOR EVALUATING THE LIKELIHOOD FUNCTION
C
      REAL*8           DET,QFORM,QNORM,PI,DNSCAL
C
C
      CHARACTER*30     SUBJID
      CHARACTER*40     HEDCOV,HEDCTM
      PARAMETER(PI=3.141592653589793238)
      DATA HEDCOV     /'ADJUSTED COVARIANCE MATRIX              '/
      DATA HEDCTM     /'CHOLESKY TRANSFORMATION MATRIX          '/
C
C
C     QMEAN  MEAN OF NORMAL APPROXIMATION TO EXAMINEE LIKELIHOOD
C            READ FROM PHASE1 WORK FILE
C
C     QVAR   VAR OF  NORMAL APPROXIMATION TO EXAMINEE LIKELIHOOD
C            READ FROM PHASE1 WORK FILE.  THE INVERSE OF THIS
C            VARIANCE IS ACTUALLY READ FROM THE PHASE1 WORK FILE.
C
C     PMEAN  PRIOR MEAN COMPUTED WITH INPUT FROM PHASE1 WORK FILE
C
C     GAMMA,SIGMA   ESTIMATES FROM THE PREVIOUS ITERATION
C
C     XMEANI,COVI   POSTERIOR MEAN AND VARIANCE FOR EACH EXAMINEE
```

```
C     COVI IS REUSED TO HOLD A DIFFERENT MATRIX IN THE
C     CALCULATION OF THE APPROXIMATE LIKELIHOOD FUNCTION.
C
C     T      CONDITIONING VARIABLES READ FROM PHASE1 WORK FILE
C
C     GWN,GMU  GROUP REPORTING VARIABLES
C
C     LKTOT  NEGATIVE TWICE THE LOG LIKELIHOOD
C
C     SINV   CONTAINS THE INVERSE OF SIGMA
C
C     TDM,COV  ACCUMULATE POSTERIOR MEANS AND VARIANCES
C
C     WMEAN  WORK SPACE FOR COMPUTING POSTERIOR MEANS
C
C
C     NOTE THAT LKTOT IS NOW A SCALAR REGARDLESS OF
C     NSCALE
C
C
C
      DNSCAL=DFLOAT(NSCALE)
      DSML = 1D-6
      DBIG = 14D1
C                                        PUT SIGMA
C                                        IN SINV AND INVERT SINV
C*VDIR: ASSUME COUNT(2)
      DO 10 JSCAL=1,NSCALE
        DO 15 ISCAL=1,NSCALE
           SINV(ISCAL,JSCAL)=SIGMA(ISCAL,JSCAL)
 15        CONTINUE
 10   CONTINUE
      CALL SYMINV(SINV,NSCALE,DET,*7000)
C
C                                        INITIALIZE LKTOT, COV & TDM
C*VDIR: ASSUME COUNT(2)
      LKTOT=0D0
      DO 40 ISCAL=1,NSCALE
C*VDIR: ASSUME COUNT(2)
         DO 20 JSCAL=1,NSCALE
            COV(JSCAL,ISCAL) = 0D0
 20      CONTINUE
C*VDIR: ASSUME COUNT(50)
         DO 30 IRANK=1,NRANK
            TDM(IRANK,ISCAL) = 0D0
 30      CONTINUE
 40   CONTINUE
C
C     INITIALIZE GROUP ACCUMULATION VARIABLES
C
C*VDIR: ASSUME COUNT(10)
```

```
      DO 60 K=1,TOTLEV
C*VDIR: ASSUME COUNT(2)
         DO 50 ISCAL=1,NSCALE
            GMU(ISCAL,K) = 0D0
 50      CONTINUE
 60   CONTINUE
      REWIND(IUNIT)
      ISUBJ = 0
C
C                                        MAIN LOOP OVER SUBJECTS
C
C
C  NESTEP  1.
C
C     READ STATEMENT MODIFIED TO READ ONLY INFO FROM
C     APPROXIMATING NORMAL FOR THE LIKELIHOOD FROM
C     READDF/READMF
C
 100  READ(IUNIT,END=500) SUBJID(1:NIDW),KUSE,WGHT,
     $             (T(N),N=1,NRANK),(QMEAN(J),J=1,NSCALE),
     $             (QVAR(J),J=1,NSCALE),
     $             (GVEC(I),I=1,NTOTG)
C
      IF (KUSE .EQ. 0) GO TO 100
C
C  NESTEP  1.  END
C
      ISUBJ = ISUBJ+1
C
      QDIAG = ICYCL .EQ. 1 .AND. ISUBJ .LE. NQC


C
C  NESTEP  2.
C
C       COMPUTE THE PRIOR MEAN FOR THETA
C
C*VDIR: ASSUME COUNT(2)
      DO 120 ISCAL=1,NSCALE
         PMI = 0D0
C*VDIR: ASSUME COUNT(50)
         DO 110 IRANK=1,NRANK
            PMI = PMI+GAMMA(IRANK,ISCAL)*T(IRANK)
 110     CONTINUE
         PMEAN(ISCAL) = PMI
 120  CONTINUE
C
```

```
C  NESTEP  2.  END
C
      IF (QDIAG) THEN
         CALL HEDOUT
         WRITE(6,8000) ISUBJ,ICYCL
         NLEFT = 55
      END IF
C
C     COMPUTE THE POSTERIOR VARIANCE OF THETA USING THE PRIOR VARIANCE
C     AND THE (DIAGONAL) VARIANCE OF THE NORMAL DISTRIBUTION
C     APPROXIMATING THE THETA-LIKELIHOOD. NOTE THAT QVAR
C     ALREADY CONTAINS THE INVERSE OF THE APPROXIMATING VARIANCE
C
C
C  NESTEP  3.
C
C
C     COMPUTE THE POSTERIOR VARIANCE OF THETA USING (4.5)
C
      DO 130 JSCAL=1,NSCALE
         DO 140 ISCAL=1,NSCALE
            COVI(ISCAL,JSCAL)=SINV(ISCAL,JSCAL)
 140     CONTINUE
         COVI(JSCAL,JSCAL)=COVI(JSCAL,JSCAL)+QVAR(JSCAL)
 130  CONTINUE
C
      CALL SYMINV(COVI,NSCALE,DET,*7000)
C
C  NESTEP  4.
C
C
C     COMPUTE THE POSTERIOR MEAN OF THETA USING (4.6)
C
      DO 150 JSCAL=1,NSCALE
         WMEAN(JSCAL)=QVAR(JSCAL)*QMEAN(JSCAL)
         DO 160 ISCAL=1,NSCALE
            WMEAN(JSCAL)=WMEAN(JSCAL)+SINV(ISCAL,JSCAL)*PMEAN(ISCAL)
 160     CONTINUE
 150  CONTINUE
C
      DO 170 JSCAL=1,NSCALE
            XMEANI(JSCAL)=0D0
         DO 180 ISCAL=1,NSCALE
            XMEANI(JSCAL)=XMEANI(JSCAL)+COVI(ISCAL,JSCAL)*WMEAN(ISCAL)
 180     CONTINUE
 170  CONTINUE
C
```

C NESTEP 4. END

```
C
C
C
      IF (QDIAG) THEN
         WRITE(6,8030)
         NLEFT = NLEFT-3
C*VDIR: ASSUME COUNT(2)
         DO 210 ISCAL=1,NSCALE
            QPVAR = 0D0
            IF (QVAR(ISCAL) .GT. 0D0) QPVAR = 1D0/QVAR(ISCAL)
            WRITE(6,8040) ISCAL,PMEAN(ISCAL),SIGMA(ISCAL,ISCAL),QPVAR,
     $                    XMEANI(ISCAL),COVI(ISCAL,ISCAL)
            NLEFT = NLEFT-1
  210    CONTINUE
      END IF
C
C
      IF (QDIAG) THEN
         CALL PRTMAT(COVI,NSCALE,CM(KSNAM),NSCALE,CM(KSNAM),NSCALE,
     $               HEDCOV)
      END IF
C
C
```

C NESTEP 5.

```
C
C        ACCUMULATE TDM AND COV WITHOUT IMPUTATIONS
C        EQUATIONS (1.7) AND (1.8)
C
C
         DO 340 JSCAL=1,NSCALE
           DO 330 IRANK=1,NRANK
              TDM(IRANK,JSCAL) = TDM(IRANK,JSCAL)+
     $                           WGHT*T(IRANK)*XMEANI(JSCAL)
 330       CONTINUE
           XMFAC=XMEANI(JSCAL)-PMEAN(JSCAL)
C
           DO 350 ISCAL=JSCAL,NSCALE
              COV(ISCAL,JSCAL) = COV(ISCAL,JSCAL)+
     $                           WGHT*COVI(ISCAL,JSCAL)
              COV(ISCAL,JSCAL) = COV(ISCAL,JSCAL)+
     $            WGHT*(XMEANI(ISCAL)-PMEAN(ISCAL))*XMFAC
              COV(JSCAL,ISCAL)=COV(ISCAL,JSCAL)
 350       CONTINUE
 340     CONTINUE
C
C    ### END OF CODE TO COMPUTE TDM AND COV
C
C
```

C NESTEP 6.

```
C
C    ###CODE TO COMPUTE THE APPROXIMATE (WEIGHTED) LOG LIKELIHOOD
C        COVI IS REUSED TO HOLD A VARIANCE MATRIX
C
      QFORM=0D0
      QNORM=(DNSCAL/2D0)*DLOG(2*PI)
      IF (NSCALE.EQ.1)THEN
         DET=SIGMA(1,1)+(1.0/MAX(DSML,QVAR(1)))
         COVI(1,1)=1.0/DET
         QNORM=QNORM+.5*DLOG(1.0/MAX(DSML,QVAR(1)))
         QFORM=( QMEAN(1)-PMEAN(1) )*( QMEAN(1)-
     $     PMEAN(1) )*COVI(1,1)
       ELSE
         DO 360 JSCAL=1,NSCALE
            QNORM=QNORM+.5*DLOG(1.0/MAX(DSML,QVAR(JSCAL)))
            DO 370 ISCAL=1,NSCALE
               COVI(ISCAL,JSCAL)=SIGMA(ISCAL,JSCAL)
 370        CONTINUE
            COVI(JSCAL,JSCAL)=COVI(JSCAL,JSCAL)+
     $                           (1.0/MAX(DSML,QVAR(JSCAL)))
 360     CONTINUE
C
         CALL SYMINV(COVI,NSCALE,DET,*7000)
         DET=DEXP(DET)
C
         DO 380 JSCAL=1,(NSCALE-1)
            QFORM=QFORM+
     $        ( QMEAN(JSCAL)-PMEAN(JSCAL) )*( QMEAN(JSCAL)-
     $         PMEAN(JSCAL) )*COVI(JSCAL,JSCAL)
            DO 390 ISCAL=(JSCAL+1),NSCALE
               QFORM=QFORM+
     $              2.0*( QMEAN(ISCAL)-PMEAN(ISCAL) )*( QMEAN(JSCAL)-
     $              PMEAN(JSCAL) )*COVI(ISCAL,JSCAL)
 390           CONTINUE
 380        CONTINUE
         QFORM=QFORM+( QMEAN(NSCALE)-PMEAN(NSCALE) )*( QMEAN(NSCALE)-
     $                PMEAN(NSCALE) )*COVI(NSCALE,NSCALE)
       END IF
C
      LKTOT=LKTOT+WGHT*(  QNORM-0.5*(LOG(DET)+QFORM)  )
C
```

C NESTEP 6. END

```
C
C
C    ### REPLACE GROUP MEAN FROM IMPUTATION ESTIMATE WITH
C        GROUP MEAN FROM ANALYTIC CALCULATION
C
      DO 440 K=1,NTOTG
```

```
         ILEV = GVEC(K)
         IF (ILEV .GT. 0) THEN
            DO 430 ISCAL=1,NSCALE
               GMU(ISCAL,ILEV) = GMU(ISCAL,ILEV)+WGHT*XMEANI(ISCAL)
 430        CONTINUE
         END IF
 440  CONTINUE
C
      GO TO 100
 500  CONTINUE
C
C
C     RETURN -2LOGLIK AND NORM WEIGHTS TO ADD TO SAMPLE SIZE
C
C
      LKTOT=-2D0*LKTOT*(DFLOAT(ISUBJ)/WGTSUM)
C
      DO 530 K=1,TOTLEV
         WT = GWN(K)
         IF (WT .NE. 0D0) THEN
            DO 520 ISCAL=1,NSCALE
               GMU(ISCAL,K) = GMU(ISCAL,K)/WT
 520        CONTINUE
         END IF
 530  CONTINUE
C
C
      RETURN
C
C
 8000 FORMAT(' <DIAG> PRINTOUT FOR SUBJECT NUMBER ',I4,
     $        '     CYCLE = ',I4)
 8010 FORMAT('0SCALE',I3,'  THETA        LK     PRIOR      POST',
     $        '    N.POST')
 8020 FORMAT(' ',I5,F10.3,4F10.5)
 8030 FORMAT('0      -----PRIOR---------           -----POSTERIOR-----'
     $ /' SCALE       MU      SIGMA   VAR(LK)      MEAN       VAR ')
 8040 FORMAT(' ',I5,5F10.5)
 8060 FORMAT('0IMPUTATION #',I4)
 8070 FORMAT('     Z: ',10F12.4)
 8080 FORMAT('     D: ',10F12.4)
C
C###  NT FIXING ALTERNATE RETURN CALLS TO SYMINV
C
 7000 WRITE(6,7010)
 7010 FORMAT(' CALL TO SYMINV FAILED IN ESTEP')
      STOP
      END
```

F Appendix Subroutine CESTEP

```
C
C     THIS ESTEP IS MODIFIED TO USE HIGHER ORDER ASYPMTOTIC
C     CORRECTIONS TO APPROXIMATE EXAMINEE POSTERIOR MEANS AND
C     VARIANCES
C
      SUBROUTINE ESTEP(COV,TDM,PMEAN,XMEANI,QMEAN,QVAR,T,
     $                 LKTOT,GAMMA,SIGMA,XSTEP,
     $                 COVI,SINV,WMEAN,D,GVEC,GMU,GWN,SLOPE,
     $                 THRESH,GUESSC,CATEG,LINK,NCAT,XITEM,
     $                 DERIV,G,XBACK,COVIB,XMODEI,ISUBJ,
     $                 IFAIL,INOTPD,IFCORM,IFCORV,AVEIT)
      IMPLICIT REAL*8 (A-H,O-Z)
      INCLUDE 'BIGMAT.FOR'
      INCLUDE 'GLOBAL.FOR'
      LOGICAL         QDIAG
      INTEGER*2       GVEC(NTOTG)
      REAL*4          T(NRANK)
      REAL*8          COV(NSCALE,NSCALE),TDM(NRANK,NSCALE),
     $                PMEAN(NSCALE),XMEANI(NSCALE),XBACK(NSCALE),
     $                QMEAN(NSCALE),QVAR(NSCALE),
     $                LKTOT,
     $                GAMMA(NRANK,NSCALE),SIGMA(NSCALE,NSCALE),
     $                XSTEP(NSCALE),
     $                COVI(NSCALE,NSCALE),SINV(NSCALE,NSCALE),
     $                WMEAN(NSCALE),D(NSCALE),GMU(NSCALE,TOTLEV),
     $                GWN(TOTLEV),DERIV(NSCALE,4),G(NSCALE,NSCALE),
     $                COVIB(NSCALE,NSCALE),XMODEI(NSCALE)
C
C
C     ITEM PARAMETERS AND RESPONSES PASSED TO CESTEP SO DERIVATIVES
C     OF THE LIKELIHOOD FUNCTION CAN BE COMPUTED DIRECTLY
C
      REAL*8 SLOPE(NITM),THRESH(NITM),GUESSC(NITM),CATEG(MAXCAT,NITM)
      REAL*4 XITEM(NITM)
      INTEGER*4 LINK(NITM),NCAT(NITM)
C
C                     VARIABLES FOR EVALUATING THE LIKELIHOOD FUNCTION
C
      REAL*8          DET,QFORM,PI,D2NSCA,DLSIG,DETG,LKNORM
C
C     LKNORM          COMPUTED IN READDF/READMF AND PASSED TO CESTEP.
C                     MAXIMUM VALUE OF THE EXAMINEE LOG LIKELIHOOD USED
C                     TO NORM THE LOG LIKELIHOOD FUNCTION SO THAT IT IS
C                     ROUGHLY COMPARABLE TO THE OTHER METHODS OF
C                     APPROXIMATING THE LOG LIKELIHOOD FUNCTION.
C
C
      CHARACTER*30    SUBJID
```

```
      CHARACTER*40    HEDCOV,HEDCTM
      PARAMETER(PI=3.141592653589793238)
      DATA HEDCOV    /'ADJUSTED COVARIANCE MATRIX              '/
      DATA HEDCTM    /'CHOLESKY TRANSFORMATION MATRIX          '/
C
C     FOR THE NORMAL APPROXIMATION THAT IS CALCULATED FIRST
C
C     QMEAN(J) AND QVAR(J) HOLD THE MEAN AND (INVERSE OF THE) VARIANCE OF
C     THE APPROXIMATING NORMAL DISTRIBUTION OF THE  EXAMINEE LIKELIHOOD
C     FUNCTION FOR SCALE J.
C
C     PMEAN HOLDS THE PRIOR MEANS
C
C     SIGMA HOLDS THE PRIOR VARIANCES (THERE IS NO PVAR).
C
C     XMEANI,COVI HOLD THE POSTERIOR MEAN AND VARIANCE.
C     COVI IS REUSED TO HOLD A DIFFERENT MATRIX IN THE
C     CALCULATION OF THE APPROXIMATE LIKELIHOOD FUNCTION.
C
C     T HOLDS THE
C     CONDITIONING VARIABLES READ FROM THE PHASE1 WORK FILE.
C
C     TDT AND COV
C     ACCUMULATE THE SUMS OF THE MEANS AND VARIANCES.
C
C     XSTEP IS MAXIMUM
C     CHANGE ALLOWED IN NEWTON-RAPHSON ITERATIONS AND MAXIMUM CHANGE
C     ALLOWED IN CORRECTED APPROXIMATIONS.
C
C     XMODEI IS THE POSTERIOR MODE.
C
C     G INVERSE OF SECOND PARTIALS.
C
C     XBACK,COVIB BACKUP OF NGROUP
C     ESTIMATES USED IN CASE CORRECTIONS FAIL.
C
C     SINV INVERSE OF SIGMA.
C
C     D IS WORK VECTOR FOR NEWTON-RAPHSON ITERATIONS.
C
C     NOTE THAT LKTOT IS NOW A SCALAR REGARDLESS OF
C     NSCALE
C
C     VARIABLES USED TO FIND MODAL VALUE OF THE POSTERIOR DISTRIBUTION
C     BY NEWTON-RAPHSON
C
      REAL*8 XIMIN,XIMAX,AVEIT,XLNLIK
      INTEGER IPD,INOTPD,IFAIL,IFCORM,IFCORV,ICFM,ICFV
C
C     FROM COMMON  XNR1,XNR2,XNR3,XNR4,XNR5,XNR6,MAXIT
C
```

```
C
C     XNR1  PROPORTION OF CHANGE (MEASURED BY SD) FROM NORMAL
C           APPROXIMATION TO THE LIKELIHOOD ALLOWED AT EACH STEP OF N-R
C
C     XNR2  STOPPING CRITERIA APPLIED TO EACH COORDINATE IN THE
C           NEWTON-RAPHSON ROUTINE
C
C     XNR3  MAGNITUDE OF THE RATIO OF THE INFO MATRIX OBTAINED AT EACH
C           N-R STEP AND THE CORRESPONDING APPROXIMATE INFORMATION FROM
C           THE NORMAL APPROXIMATION TO THE LIKELIHOOD FUNCTION.
C           THIS RATIO IS USED TO INSURE A POSITIVE DEFINITE MATRIX
C           IN THE NEWTON RAPHSON ROUTINE
C
C     XNR4  NUMBER OF XSTEPS OF TOTAL CHANGE ALLOWED IN XMEANI
C
C     XNR5  MAXIMUM RATIO OF MODAL VARIANCE ESTIMATE TO NORMAL
C           BASED ESTIMATE ALLOWED
C
C     XNR6  MINIMUM RATIO OF CORRECTED VARIANCES TO NORMAL BASED VARIANCES
C
C     XIMIN,XIMAX  MINIMUM AND MAXIMUM DIAGONAL ELEMENTS ALONG DIAGONAL
C           OF THE NORMAL BASED INFORMATION MATRIX
C
C     XLNLIK EXAMAINEE LOG LIKELIHOOD FUNCTION EVALUATED AT
C            THE CONVERGED VALUE OF THE NEWTON ROUTINE FOR USE
C            IN COMPUTING THE LIKELIHOOD FUNCTION
C
C     MAXIT  MAXIMUM NUMBER OF ITERATIONS ALLOWED
C
C     DERIV  HOLDS THE FIRST 4 DERIVATIVES OF THE
C            NEGATIVE LOGLIKELIHOOD FUNCTION
C
C     G      HOLDS SECOND DERIVATIVES OF THE NEGATIVE LOG POSTERIOR
C
C     INOTPD COUNTS THE NUMBER OF TIMES THE INFO MATRIX IS NOT WELL
C     IPD    CONDITIONED
C
C     XNLIK  THETA LOG LIKELIHOOD
C
C     AVEIT   AVERAGE NUMBER OF ITERATIONS FOR THE NEWTON SEARCH
C
C     IFAIL   COUNTS THE NUMBER OF NEWTON SEARCH FAILURES
C
C     IFCORM  COUNTS THE NUMBER OF TIMES NO CORRECTION TO THE MEAN WAS
C     ICFM    APPLIED(WHEN THE NEWTON WAS SUCCESSFUL)
C
C     IFCORV  COUNTS THE NUMBER OF TIMES NO CORRECTION
C     ICFV    TO THE VARIANCE WAS APPLIED(WHEN THE
C                  NEWTON WAS SUCCESSFUL)
C
C
```

```
      IFAIL=0
      IFCORM=0
      IFCORV=0
      INOTPD=0
      AVEIT=0D0
C
C
      D2NSCA=DFLOAT(NSCALE)/(2.0D0)
      DSML = 1D-6
      DBIG = 14D1
C
C                                     PUT SIGMA
C                                     IN SINV AND INVERT SINV
C*VDIR: ASSUME COUNT(2)
      DO 10 JSCAL=1,NSCALE
        DO 15 ISCAL=1,NSCALE
           SINV(ISCAL,JSCAL)=SIGMA(ISCAL,JSCAL)
 15        CONTINUE
 10   CONTINUE
      CALL SYMINV(SINV,NSCALE,DLSIG,*7000)
C
C                                     INITIALIZE LKTOT, COV & TDM
C*VDIR: ASSUME COUNT(2)
      LKTOT=0D0
      DO 40 ISCAL=1,NSCALE
C*VDIR: ASSUME COUNT(2)
         DO 20 JSCAL=1,NSCALE
            COV(JSCAL,ISCAL) = 0D0
 20      CONTINUE
C*VDIR: ASSUME COUNT(50)
         DO 30 IRANK=1,NRANK
            TDM(IRANK,ISCAL) = 0D0
 30      CONTINUE
 40   CONTINUE
C
C     INITIALIZE GROUP ACCUMULATION VARIABLES
C
C*VDIR: ASSUME COUNT(10)
      DO 60 K=1,TOTLEV
C*VDIR: ASSUME COUNT(2)
         DO 50 ISCAL=1,NSCALE
            GMU(ISCAL,K) = 0D0
 50      CONTINUE
 60   CONTINUE
      REWIND(IUNIT)
      ISUBJ = 0
C
C                                     MAIN LOOP OVER SUBJECTS
C
```

C CESTEP 1.

```
C
C
C     READ STATEMENT MODIFIED TO READ ONLY INFO FROM
C     APPROXIMATING NORMAL FOR THE LIKELIHOOD FROM
C     READDF/READMF.  IN ADDITION, THE CORRECTED NORMAL
C     READS THE ITEM RESPONSES FOR COMPUTATION OF DERIVATIVES
C     OF THE LOG LIKELIHOOD FUNCTION
C
 100  READ(IUNIT,END=500) SUBJID(1:NIDW),KUSE,WGHT,
     $             (T(N),N=1,NRANK),(QMEAN(J),J=1,NSCALE),
     $             (QVAR(J),J=1,NSCALE),LKNORM,
     $             (GVEC(I),I=1,NTOTG),(XITEM(ITM),ITM=1,NITM)
C
      IF (KUSE .EQ. 0) GO TO 100
C
C  CESTEP  1.  END
C
      ISUBJ = ISUBJ+1
C
      QDIAG = ICYCL .EQ. 1 .AND. ISUBJ .LE. NQC
C
C
C
C  CESTEP  2.
C
C       COMPUTE THE PRIOR MEAN FOR THETA
C
C*VDIR: ASSUME COUNT(2)
      DO 120 ISCAL=1,NSCALE
         PMI = 0D0
C*VDIR: ASSUME COUNT(50)
         DO 110 IRANK=1,NRANK
            PMI = PMI+GAMMA(IRANK,ISCAL)*T(IRANK)
 110     CONTINUE
         PMEAN(ISCAL) = PMI
 120  CONTINUE
C
C  CESTEP 2.  END
C
      IF (QDIAG) THEN
         CALL HEDOUT
         WRITE(6,8000) ISUBJ,ICYCL
         NLEFT = 55
      END IF
C
C
C
```

```
C  CESTEP  3.
C
C
C     USE NORMAL APPROXIMATION FROM 1990 TECHNICAL REPORT
C     TO OBTAIN INITIAL STARTING VALUE TO OBTAIN THE MODE
C     OF THETA
C
C
C     COMPUTE THE POSTERIOR VARIANCE OF THETA USING THE PRIOR VARIANCE
C     AND THE (DIAGONAL) VARIANCE OF THE NORMAL DISTRIBUTION
C     APPROXIMATING THE EXAMINEE LIKELIHOOD. NOTE THAT QVAR
C     ALREADY CONTAINS THE INVERSE OF THE APPROXIMATING VARIANCE
C
C
      DO 130 JSCAL=1,NSCALE
         DO 140 ISCAL=1,NSCALE
            COVI(ISCAL,JSCAL)=SINV(ISCAL,JSCAL)
 140     CONTINUE
         COVI(JSCAL,JSCAL)=COVI(JSCAL,JSCAL)+QVAR(JSCAL)
 130  CONTINUE
C
C
C     COMPUTE THE MAX AND MIN DIAGONAL ELEMENTS.  THESE BOUNDS ARE
C     USED BY THE CODE TO CORRECT THE NORMAL APPROXIMATION.
C
C
      XIMIN=COVI(1,1)
      XIMAX=XIMIN
      DO 2010 JSCAL=2,NSCALE
         IF( COVI(JSCAL,JSCAL).GT.XIMAX )XIMAX=COVI(JSCAL,JSCAL)
         IF( COVI(JSCAL,JSCAL).LT.XIMIN )XIMIN=COVI(JSCAL,JSCAL)
 2010 CONTINUE
C
C
C
      CALL SYMINV(COVI,NSCALE,DET,*7000)
C
C     COMPUTE THE POSTERIOR MEAN OF THETA
C
      DO 150 JSCAL=1,NSCALE
         WMEAN(JSCAL)=QVAR(JSCAL)*QMEAN(JSCAL)
         DO 160 ISCAL=1,NSCALE
            WMEAN(JSCAL)=WMEAN(JSCAL)+SINV(ISCAL,JSCAL)*PMEAN(ISCAL)
 160     CONTINUE
 150  CONTINUE
C
      DO 170 JSCAL=1,NSCALE
            XMEANI(JSCAL)=0D0
         DO 180 ISCAL=1,NSCALE
            XMEANI(JSCAL)=XMEANI(JSCAL)+COVI(ISCAL,JSCAL)*WMEAN(ISCAL)
 180     CONTINUE
```

```
 170  CONTINUE
C
C  CESTEP  3.  END
C
C
C*********************************************************************
C NEWTON-RAPHSON ROUTINE TO FIND THE MODAL VALUE OF THE POSTERIOR
C
C     WMEAN REUSED AS A WORK VECTOR
C
C
C  CESTEP  4.
C
C     COMPUTE THE MAX STEP SIZES ALLOWED FOR EACH COORDINATE.
C     MAKE BACKUP COPY OF POSTERIOR MEANS AND VARIANCES FROM THE NORMAL
C     APPROXIMATION IN CASE THE CORRECTION FAILS.
C
C     THE STARTING VALUES OF NEGATIVE MEANS ARE INCREASED
C     BECAUSE THE DISTRIBUTIONS ARE TYPICALLY SKEWED WITH
C     THE MEAN MORE NEGATIVE THAN THE MODE
C
      DO 2020 JSCAL=1,NSCALE
         XSTEP(JSCAL)=XNR1*DSQRT(COVI(JSCAL,JSCAL))
         XBACK(JSCAL)=XMEANI(JSCAL)
         XMODEI(JSCAL)=XMEANI(JSCAL)
         IF(XMODEI(JSCAL).LT.0D0)
     $        XMODEI(JSCAL)=XMODEI(JSCAL)+0.5D0*XSTEP(JSCAL)
         COVIB(JSCAL,JSCAL)=COVI(JSCAL,JSCAL)
         DO 2025 ISCAL=(JSCAL+1),NSCALE
            COVIB(ISCAL,JSCAL)=COVI(ISCAL,JSCAL)
            COVIB(JSCAL,ISCAL)=COVI(JSCAL,ISCAL)
 2025    CONTINUE
 2020 CONTINUE
C
C  CESTEP  4.  END
C
C
C
C
C     SET CONSTANTS FOR ENSURING A WELL CONDITIONED INFO MATRIX
C
      B2=XNR3*XIMAX
      DMIN=XIMIN/XNR3
C
C  CESTEP  5.
C
```

```
C      BEGIN N-R ITERATIONS
C
      IPD=0
      ITER=0
 2100    ITER=ITER+1
C
C  CESTEP  5(A)
C
C        COMPUTE THE DERIVATIVES OF THE LOG LIKELIHOOD FUNCTION
C
         CALL DERVS2(XITEM,NITM,NSCALE,MAXCAT,SLOPE,THRESH,
     $              GUESSC,CATEG,LINK,NCAT,XMODEI,DERIV,DFAC)
C
C        COMPUTE THE DERIVATIVES OF THE NEGATIVE LOG POSTERIOR
C
         DO 2210 JSCAL=1,NSCALE
            DERIV(JSCAL,1)=DERIV(JSCAL,1)
            DO 2220 ISCAL=1,NSCALE
               G(ISCAL,JSCAL)=SINV(ISCAL,JSCAL)
               DERIV(JSCAL,1)=DERIV(JSCAL,1)+
     $              SINV(ISCAL,JSCAL)*( XMODEI(ISCAL)-PMEAN(ISCAL) )
 2220       CONTINUE
            G(JSCAL,JSCAL)=G(JSCAL,JSCAL)+DERIV(JSCAL,2)
 2210    CONTINUE
C
C CESTEP  5(B)
C
C        COMPUTE THE CHOLESKY DECOMPOSITION OF G
C
         CALL DCHOL(G,NSCALE,NSCALE,INFO,B2,DMIN)
         IPD=IPD+INFO
C
C        COMPUTE G^(-1)%*(FIRST DERIVATIVES)
C
         CALL CHOLMU(G,DERIV(1,1),NSCALE,WMEAN,D)
C
C  CESTEP  5(C)
C
C        UPDATE THE ESTIMATE OF THE POSTERIOR MODE AND CHECK FOR
C        CONVERGENCE
C
         ICONV=1
         DO 2230 JSCAL=1,NSCALE
            IF( DABS(D(JSCAL)).GT.XSTEP(JSCAL) )
     $           D(JSCAL)=SIGN( XSTEP(JSCAL),D(JSCAL) )
            IF( DABS(D(JSCAL)).GT.XNR2 )ICONV=0
            XMODEI(JSCAL)=XMODEI(JSCAL)-D(JSCAL)
```

```
 2230    CONTINUE
C
      IF( (ITER.LT.MAXIT).AND.(ICONV.EQ.0) ) GO TO 2100
C
C     COUNT AND REPORT THE AVERAGE NUMBER OF ITERATIONS REQUIRED
C
      AVEIT=AVEIT+ITER
C
C     COUNT THE NUMBER OF SUBJECTS THAT HAD A NON POSITIVE
C     DEFINITE MATRIX DURING SOME ITERATION
C
      IF(IPD.GT.0)INOTPD=INOTPD+1
C
C                  END OF N-R ITERATION
C*********************************************************************
C
C  CESTEP  5.  END
C
C*********************************************************************
C
C  CESTEP  6.
C
C     APPLY HIGHER ORDER ASYMPTOTIC CORRECTIONS.
C     IF THE N-R CODE FAILED TO ACHIEVE CONVERGENCE, THEN DO
C     NOT PERFORM ADJUSTMENT, USE THE ORGINAL VALUES FROM THE
C     NORMAL APPPROXIMATION TO THE LIKELIHOOD FUNCTION.
C
      ICFM=0
      ICFV=0
      IF(ICONV.EQ.1)THEN
C
C        COMPUTE THE INVERSE OF THE FINAL INFORMATION MATRIX FROM
C        THE N-R ROUTINE.  THE VARIANCES THIS YIELDS ARE USED IN THE
C        HIGHER ORDER CORRECTION TERMS.  THE DETERMINANT OF THE
C        ASYMPTOTIC VARIANCE MATRIX IS COMPUTED FOR USE IN THE
C        CALCULATION OF THE LIKELIHOOD FUNCTION
C
C  CESTEP  6(A)
C
         CALL CHOLIN(G,NSCALE,DETG)
C
C        CHOLIN RETURNS THE DETERMINANT OF G BEFORE INVERSION.
C
         DETG=1D0/DETG
C
C
C        FIRST COMPUTE THE HIGHER (3 AND 4)
```

```
C       DERIVATIVES OF THE -LOG POSTERIOR(=DERVS OF -LOG LIK) AT
C       THE FINAL ESTIMATE OF THE MODE FROM THE N-R.
C       THE FIRST TWO DERIVATIVES ARE REUSED FROM THE
C       N-R CALCULATIONS.  THEY ARE NOT RECOMPUTED HERE.
C
        CALL DERVS(XITEM,NITM,NSCALE,MAXCAT,SLOPE,THRESH,GUESSC,
     $                  CATEG,LINK,NCAT,XMODEI,DERIV,DFAC,XLNLIK)
C
C  CESTEP  6(B)
C
        DO 2350 JSCAL=1,NSCALE
C
C          FIRST ORDER CORRECTION TO THE MEANS IN (3.3)
C
           W=0D0
           DO 2375 ISCAL=1,NSCALE
              W=W+G(ISCAL,ISCAL)*G(ISCAL,JSCAL)*DERIV(ISCAL,3)
C
 2375      CONTINUE
           W=0.5D0*W
C
           XMEANI(JSCAL)=XMODEI(JSCAL)-W
C
C  CESTEP  6(C)
C
C          CHECK SIZE OF THE CORRECTION TO THE MEAN
C          BOUND AMOUNT OF CHANGE ALLOWED
C          XS AND XD ARE LOCAL WORK VARIABLES
C
           XS=XNR4*XSTEP(JSCAL)
           XD=XMEANI(JSCAL)-XBACK(JSCAL)
        IF( DABS(XD) .GT. XS)THEN
              ICFM=1
              XMEANI(JSCAL)=XBACK(JSCAL)+SIGN(XS,XD)
           END IF
C
C
C
C  CESTEP  6(D)
C
C          AVERAGE THE TWO VARIANCE ESTIMATES FOR
C          BOUND CHECK. THE WEIGHTING OF THE AVERAGE
C          WAS DETERMINED EMPIRICALLY FROM A NUMBER OF EXAMPLES.
C          IF THE MODAL VARIANCE ESTIMATE IS TOO LARGE, THEN
C          JUST USE THE NORMAL-BASED ESTIMATE
C
           IF( G(JSCAL,JSCAL)/COVIB(JSCAL,JSCAL) .GT. XNR5)THEN
```

```
            W=COVIB(JSCAL,JSCAL)
         ELSE
            W=0.25D0*COVIB(JSCAL,JSCAL)+0.75D0*G(JSCAL,JSCAL)
         END IF
C
         DO 2385 ISCAL=JSCAL,NSCALE
C
```

C CESTEP 6(E)

```
C
C           APPLY SECOND ORDER CORRECTION TO THE VARIANCES IN (3.4)
C
C
            W3=0D0
            W3A=0D0
            W4=0D0
            DO 2410 K=1,NSCALE
               W4=W4+DERIV(K,4)*G(K,ISCAL)*G(K,JSCAL)*G(K,K)
               W3A=W3A+DERIV(K,3)*DERIV(K,3)*G(K,ISCAL)*
     $                              G(K,JSCAL)*G(K,K)*G(K,K)
               W3B=0D0
               DO 2420 L=1,(K-1)
                  W3B=W3B+DERIV(L,3)*G(L,K)*
     $               ( G(L,ISCAL)*G(L,JSCAL)*G(K,K) +
     $                 G(L,ISCAL)*G(K,JSCAL)*G(L,K) +
     $                    G(K,ISCAL)*G(L,JSCAL)*G(L,K) +
     $                    G(K,ISCAL)*G(K,JSCAL)*G(L,L) )
 2420             CONTINUE
               W3=W3+W3B*DERIV(K,3)
 2410          CONTINUE
               COVI(ISCAL,JSCAL)=G(ISCAL,JSCAL)-0.5D0*W4+W3A+0.5D0*W3
               COVI(JSCAL,ISCAL)=COVI(ISCAL,JSCAL)
 2385       CONTINUE
C
```

C CESTEP 6(F)

```
C
C           BOUND THE MAGNITUDE OF THE CORRECTION
C
            IF(COVI(JSCAL,JSCAL)/W .GT. XNR5)THEN
               ICFV=1
               COVI(JSCAL,JSCAL)=W*XNR5
            ELSE IF(COVI(JSCAL,JSCAL).LE.W*XNR6)THEN
               ICFV=1
               COVI(JSCAL,JSCAL)=W*XNR6
            END IF
C
C
 2350    CONTINUE
C
```

```
C        USE CORRELATIONS FROM THE ORIGINAL NORMAL APPROXIMATION
C        IF THE CORRECTION FAILED ON ANY VARIANCE
C
         IF(ICFM.EQ.1)IFCORM=IFCORM+1
C
C  CESTEP  6(G)
C
         IF(ICFV.EQ.1)THEN
            IFCORV=IFCORV+1
            DO 2310 JSCAL=1,NSCALE
               DO 2305 ISCAL=(JSCAL+1),NSCALE
                  XSD=DSQRT(COVI(JSCAL,JSCAL)*COVI(ISCAL,ISCAL))
                  XBSD=DSQRT(COVIB(JSCAL,JSCAL)*COVIB(ISCAL,ISCAL))
                  XBCOR=COVIB(ISCAL,JSCAL)/XBSD
                  COVI(ISCAL,JSCAL)=XSD*XBCOR
                  COVI(JSCAL,ISCAL)=COVI(ISCAL,JSCAL)
 2305          CONTINUE
 2310       CONTINUE
         END IF
C
C  CESTEP  6(G)  END
C
C
C
C     NEWTON-RAPHSHON FAILED TO CONVERGE, USE NORMAL-BASED MOMENTS
C     ALREADY STORED IN XMEANI AND COVI
C
      ELSE
          IFAIL=IFAIL+1
      END IF
C
C
C
      IF (QDIAG) THEN
         WRITE(6,8030)
         NLEFT = NLEFT-3
C*VDIR: ASSUME COUNT(2)
         DO 210 ISCAL=1,NSCALE
            QPVAR = 0D0
            IF (QVAR(ISCAL) .GT. 0D0) QPVAR = 1D0/QVAR(ISCAL)
            WRITE(6,8040) ISCAL,PMEAN(ISCAL),SIGMA(ISCAL,ISCAL),QPVAR,
     $                    XMEANI(ISCAL),COVI(ISCAL,ISCAL)
            NLEFT = NLEFT-1
  210    CONTINUE
      END IF
C
C
      IF (QDIAG) THEN
```


63

```
      CALL PRTMAT(COVI,NSCALE,CM(KSNAM),NSCALE,CM(KSNAM),NSCALE,
     $               HEDCOV)
      END IF
C
C
C  CESTEP  7.
C
C        ACCUMULATE TDM AND COV WITHOUT IMPUTATIONS
C        EQUATIONS (1.7) AND (1.8)
C
C
         DO 340 JSCAL=1,NSCALE
           DO 330 IRANK=1,NRANK
              TDM(IRANK,JSCAL) = TDM(IRANK,JSCAL)+
     $                           WGHT*T(IRANK)*XMEANI(JSCAL)
 330       CONTINUE
           XMFAC=XMEANI(JSCAL)-PMEAN(JSCAL)
C
           DO 350 ISCAL=JSCAL,NSCALE
              COV(ISCAL,JSCAL) = COV(ISCAL,JSCAL)+
     $                            WGHT*COVI(ISCAL,JSCAL)
              COV(ISCAL,JSCAL) = COV(ISCAL,JSCAL)+
     $            WGHT*(XMEANI(ISCAL)-PMEAN(ISCAL))*XMFAC
              COV(JSCAL,ISCAL)=COV(ISCAL,JSCAL)
 350       CONTINUE
 340     CONTINUE
C
C    ### END OF CODE TO COMPUTE TDM AND COV
C
C
C
C  CESTEP  8.
C
C
C    ### CODE TO COMPUTE THE APPROXIMATE (WEIGHTED) LOG LIKELIHOOD
C        COVI IS REUSED TO HOLD A VARIANCE MATRIX
C
C  CESTEP  8(A)
C
C
C     USE NORMAL APPROXIMATION VALUES IF MODE IS NOT FOUND
C
      IF( (ICONV .EQ. 0) )THEN
         QFORM=0.0D0
         DO 382 JSCAL=1,NSCALE
            QFORM=QFORM+( XMEANI(JSCAL)-PMEAN(JSCAL) )*( XMEANI(JSCAL)-
     $           PMEAN(JSCAL) )*SINV(JSCAL,JSCAL)
            QJDIFF=2.0*( XMEANI(JSCAL)-PMEAN(JSCAL) )
            DO 384 ISCAL=(JSCAL+1),NSCALE
```

```
            QFORM=QFORM+QJDIFF*( XMEANI(ISCAL)-
     $                  PMEAN(ISCAL) )*SINV(ISCAL,JSCAL)
 384       CONTINUE
 382     CONTINUE
         LKTOT=LKTOT+ WGHT*( (XLNLIK-LKNORM)+0.5D0*( DLOG(DETG)-DLSIG )
     $          -0.5D0*QFORM + D2NSCA*DLOG(2.0D0*PI) )
C
      ELSE
C
C  CESTEP  8(B)
C
C        APPLY CORRECTION IN (??) ACCURATE UP TO N^(-1/2) ONLY
C
C
         QFORM=0.0D0
         DO 392 JSCAL=1,NSCALE
            QFORM=QFORM+( XMODEI(JSCAL)-PMEAN(JSCAL) )*( XMODEI(JSCAL)-
     $            PMEAN(JSCAL) )*SINV(JSCAL,JSCAL)
            QJDIFF=2.0*( XMODEI(JSCAL)-PMEAN(JSCAL) )
            DO 394 ISCAL=(JSCAL+1),NSCALE
               QFORM=QFORM+QJDIFF*( XMODEI(ISCAL)-
     $                  PMEAN(ISCAL) )*SINV(ISCAL,JSCAL)
 394        CONTINUE
 392     CONTINUE
         LKTOT=LKTOT+ WGHT*( (XLNLIK-LKNORM)+0.5D0*( DLOG(DETG)-DLSIG )
     $          -0.5D0*QFORM + D2NSCA*DLOG(2.0D0*PI) )
      END IF
C
C  CESTEP  8.  END
C
C
C     ### REPLACE GROUP MEAN FROM IMPUTATION ESTIMATE WITH
C         GROUP MEAN FROM ANALYTIC CALCULATION
C
      DO 440 K=1,NTOTG
         ILEV = GVEC(K)
         IF (ILEV .GT. 0) THEN
            DO 430 ISCAL=1,NSCALE
               GMU(ISCAL,ILEV) = GMU(ISCAL,ILEV)+WGHT*XMEANI(ISCAL)
 430        CONTINUE
         END IF
 440  CONTINUE
C
      GO TO 100
 500  CONTINUE
C
C
C     RETURN -2LOGLIK AND NORM WEIGHTS TO ADD TO SAMPLE SIZE
```

```
C
C
      LKTOT=-2D0*LKTOT*(DFLOAT(ISUBJ)/WGTSUM)
C
      AVEIT=AVEIT/DFLOAT(ISUBJ)
C
      DO 530 K=1,TOTLEV
         WT = GWN(K)
         IF (WT .NE. 0D0) THEN
            DO 520 ISCAL=1,NSCALE
               GMU(ISCAL,K) = GMU(ISCAL,K)/WT
 520        CONTINUE
         END IF
 530  CONTINUE
C
      RETURN
C
C
 8000 FORMAT(' <DIAG> PRINTOUT FOR SUBJECT NUMBER ',I4,
     $       '     CYCLE = ',I4)
 8010 FORMAT('0SCALE',I3,'  THETA        LK     PRIOR      POST',
     $       '    N.POST')
 8020 FORMAT(' ',I5,F10.3,4F10.5)
 8030 FORMAT('0      -----PRIOR---------           -----POSTERIOR-----'
     $ /' SCALE       MU      SIGMA   VAR(LK)      MEAN       VAR ')
 8040 FORMAT(' ',I5,5F10.5)
C
C###  NT FIXING ALTERNATE RETURN CALLS TO SYMINV
C
 7000 WRITE(6,7010)
 7010 FORMAT(' CALL TO SYMINV FAILED IN ESTEP')
      STOP
      END
```

REFERENCES

Beaton, A. (1987). *The NAEP 1983-1984 Technical Report*, NAEP Report 15-TR-20. Princeton, NJ: Educational Testing Service.

Beaton, A., and Zwick, R. (1992). "Overview of the National Assessment of Educational Progress", *Journal of Educational Statistics*, **17**, 95-109.

Bock, R. D., and Aitkin, M. (1981). "Marginal maximum likelihood estimation of item parameters: application of an EM ", *Psychometrika*, **46**, 443-459.

Box, E.P., and Tiao, G.(1973). *Bayesian inference in statistical analysis*, Reading, Mass: Addison-Wesley.

de Bruijn, N. G. (1981). *Asymptotic methods in analysis*, New York: Dover.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, **39**, 1-38.

Haberman, S. (1977). "Maximum likelihood estimation in exponential response models", Annals of Statistics, **5**, 815-841.

Hambleton, R., and Swaminathan, H. (1985). *Item response theory: principles and applications*, Boston: Kluwer-Nijhoff.

Johnson, E. G., and Zwick, R. (1990). *Focusing the new design: The NAEP 1988 Technical Report*, NAEP Report 19-TR-20. Princeton, NJ: Educational Testing Service.

Kass, R., and Steffey, D. (1989). "Approximate bayesian inference in conditionally independent hierarchical models (parametric empirical bayes models)", *J. Am. Statist. Assoc.*, **84**, 717-726.

Kennedy, W., and Gentleman, J. (1980). *Statistical Computing*, New York: Marcel Dekker.

Kirsch, I., and Jungeblut, A. (1986). *Literacy: Profiles of America's young adults*, NAEP Report 16-OL-01. Princeton, NJ: Educational Testing Service.

Little, R., and Rubin, D. (1987). *Statistical analysis with missing data*, New York: Wiley.

Mislevy, R. (1984). "Estimating latent distributions", *Psychometrika*, **44**, 358-381.

Mislevy, R. (1985). "Estimation of latent group effects", *J. Am. Statist. Assoc.*, **80**, 993-997.

Mislevy, R. (1990). "Scaling procedures", in *Focusing the new design: The NAEP 1988Technical Report*, Ed. Johnson, E.G., and Zwick, R., NAEP Report 19-TR-20. Princeton, NJ: Educational Testing Service.

Mislevy, R. (1991). "Randomization-based inference about latent variables from complex samples", *Psychometrika*, **56**, 177-196.

Mislevy, R., Johnson, E. G., Muraki, E. (1992). "Scaling procedures in NAEP", *Journal of Educational Statistics*, **17**, 131-154.

Mislevy, R.J., and Sheehan, K.M.(1987). "Marginal estimation procedures", In A.E. Beaton, *The NAEP 1983/84 Technical Report*, NAEP Report 15-TR-20. Princeton: Educational Testing Service.

Mosteller, F., and Wallace, D. (1964). *Applied bayesian and classical inference: the case of the Federalist Papers*, New York: Springer-Verlag.

Muraki, E. (1992). "A generalized partial credit model: application of an EM algorithm", *Applied Psychological Measurement*, **16**, 159-176.

Tierney, L., Kass, R., and Kadane, J. (1989). "Fully exponential Laplace approximations to expectations and variances of nonpositive functions", *J. Am. Statist. Assoc.*, **84**, 710-716.

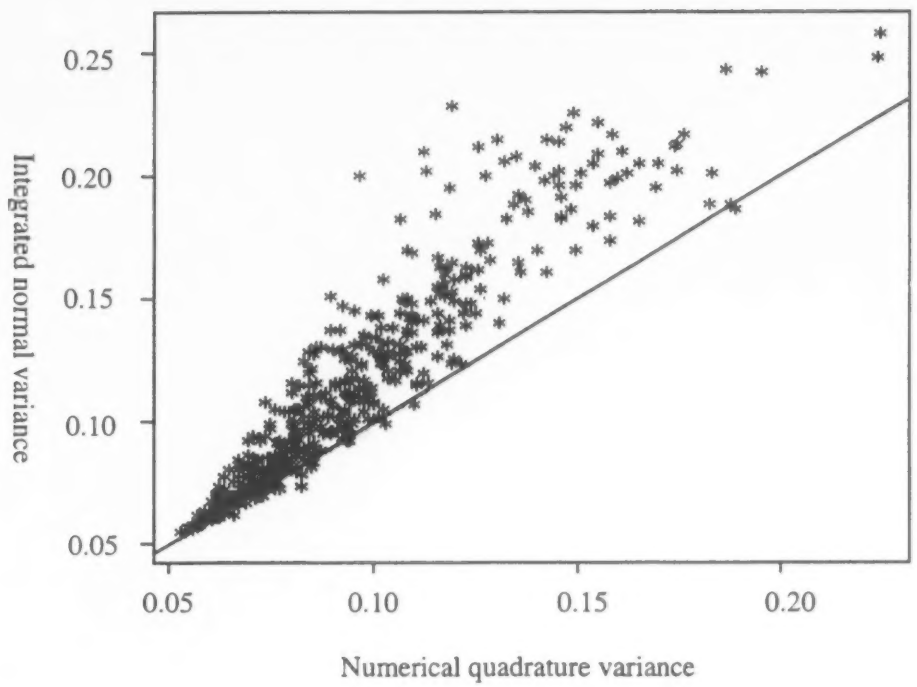

Figure 1: Examinee posterior variances for the measurement scale computed using the non-modal normal approximation versus the examinee posterior variances computed using numerical quadrature with the same values of the parameters input to both procedures.

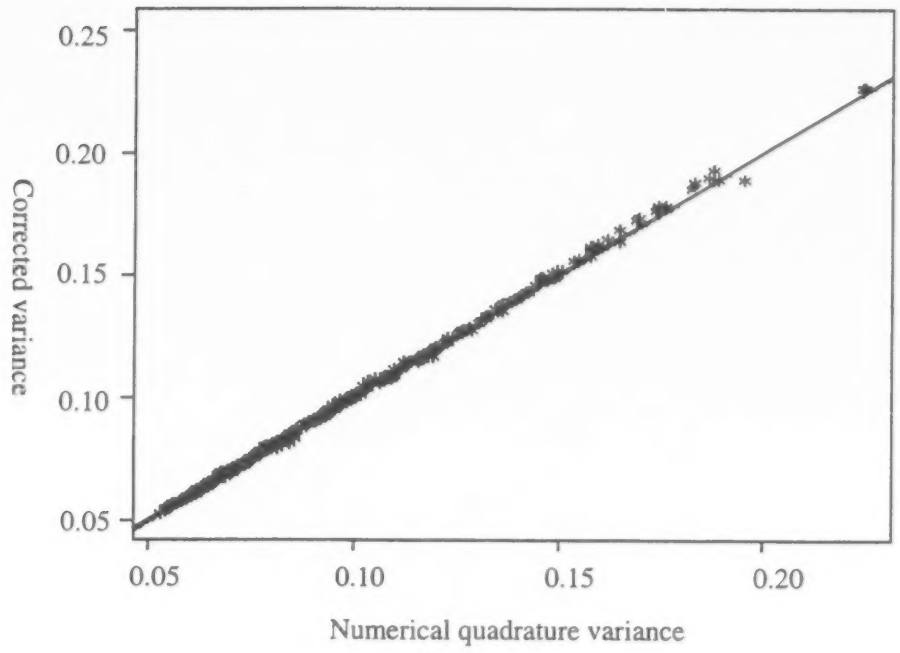

Figure 2: Examinee posterior variances for the measurement scale computed using the asymptotic corrections versus the examinee posterior variances computed using numerical quadrature with the same values of the parameters input to both procedures.